

Faculdade de Engenharia da Universidade do Porto



Agente de contexto para dispositivos móveis

Alexandre Manuel de Azevedo Maia Ribeiro Fernandes

Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores
Major Telecomunicações

Orientador: Prof^a Maria Teresa Andrade

6 de Fevereiro de 2009

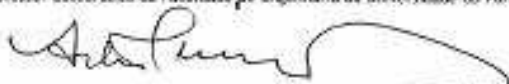
A Dissertação intitulada

"AGENTE DE CONTEXTO PARA DISPOSITIVOS MÓVEIS"

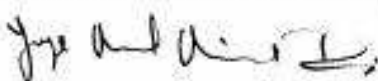
foi aprovada em provas realizadas em 04/Março/2009

o júri

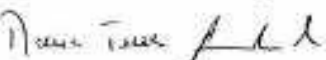
Presidente Professor Doutor Artur Pimenta Alves
Professor Catedrático da Faculdade de Engenharia da Universidade do Porto



Professor Doutor Jorge Manuel Miranda Dias
Professor Associado da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

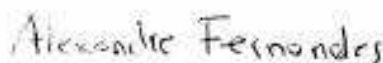


Professor Doutor Maria Teresa Magalhães da Silva Pinto de Andrade
Professora Auxiliar da Faculdade de Engenharia da Universidade do Porto



O autor declara que a presente dissertação (ou relatório de projecto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extractos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são correctamente citados.

Autor - ALEXANDRE MANUEL DE AZEVEDO MAIA RIBEIRO FERNANDES



Faculdade de Engenharia da Universidade do Porto

© Alexandre Fernandes, 2009

Resumo

Neste trabalho é desenvolvido um sistema de extracção automática das capacidades de dispositivos móveis com vista a auxiliar e a adequar a entrega de conteúdo. O sistema está implementado num servidor aplicacional juntamente com um repositório de informação dos dispositivos. Todas as características do dispositivo são entregues ao cliente num formato adequado à sua correcta visualização. A identificação correcta do dispositivo, a abrangência da heterogeneidade deste tipo de clientes e a disponibilização da pluralidade das capacidades é a sua característica de maior relevância. São também consideradas as características dinâmicas da operação decorrente do uso de um dispositivo móvel com vista a dotar o sistema de contexto mais abrangente numa implementação futura. São utilizadas as ferramentas WURFL e WALL e a linguagem Java para implementação.

Página em branco

Abstract

This work developed a system for automatic extraction capabilities of mobile devices to assist and tailor the delivery of content. The system is implemented within an application server which contains a repository of information of the devices. All features are delivered to the client device in a format appropriate to their proper display. The correct identification of the device, the extent of the heterogeneity of this type of clients and providing the plurality of capabilities are the system's major characteristics. The dynamic characteristics inherent to the mode of operation of a mobile device are also considered in order to give the system more comprehensive context on a future implementation. It has been developed with the tools WURFL and WALL for Java implementation.

Página em branco

Página em branco

Índice

Capítulo 1	16
Introdução	16
1.2 - Definição de Contexto	18
1.3 - Context-Aware.....	18
1.4 - O agente	19
Capítulo 2	21
Estado da Arte.....	21
2.1 Detecção de dispositivos	22
2.2 Adaptação de conteúdo.....	24
2.3 WURFL	29
2.4 WALL	36
2.5 UAPROF	37
Capítulo 3	44
Arquitectura	44
Capítulo 4	48
4.1 Implementação.....	48
4.2 Resultados obtidos.....	54
Capítulo 5	66
5.1 Conclusões	66
5.2 Perspectivas de desenvolvimento	67
Anexos	68

Lista de figuras

Figura 1. - Intersecção das linguagens de markup de dispositivos móveis.

Figura 2. - Classes da API Java do WURFL implementadas numa aplicação.

Figura 3. - Camadas de uma aplicação que utiliza a biblioteca WALL.

Figura 4. - Arquitectura proposta do sistema.

Figura 5. - Configuração do ficheiro web.xml.

Figuras 6. ; 15. ; 22. ; 23. - Identificação do dispositivo móvel/não móvel.

Figuras 7. ; 8. ; 9. ; 10. ; 16. ; 17. ; 18. - Hardware.

Figuras 11. ; 12. ; 13. ; 14. ; 19. ; 20. ; 21. - Software.

Abreviaturas e Símbolos

Lista de abreviaturas

cHTML	<i>Compact Hyper Text Markup Language</i>
DEEC	Departamento de Engenharia Electrotécnica e de Computadores
FEUP	Faculdade de Engenharia da Universidade do Porto
JEE	<i>Java Enterprise Edition</i>
JME	<i>Java Micro Edition</i>
PDA	<i>Personal Digital Assistant</i>
W3C	<i>World Wide Web Consortium</i>
WALL	<i>Wireless Abstraction Library</i>
WML	<i>Wireless Markup Language</i>
WURFL	<i>Wireless Universal Resource File</i>
XHTML-MP	<i>eXtended Hyper Text Markup Language Mobile Profile</i>

Capítulo 1

Introdução

A rápida e crescente evolução da área de computação móvel introduziu novos desafios no desenvolvimento de aplicações e prestação de serviços. Esta área é caracterizada pelo estudo de dispositivos móveis, suas aplicações e a consequente integração em serviços e redes existentes. Entre estes dispositivos incluem-se o telemóvel ou smartphone, o PDA, o pager, o notebook, etc. A integração deste tipo de dispositivos em serviços já existentes visa essencialmente a adaptação dos serviços a um ambiente de mobilidade do utilizador sem que a qualidade oferecida sofra uma degradação considerável. Para atingir o seu propósito, o desenvolvimento destes serviços tem que ter em consideração a grande variedade dos dispositivos existentes juntamente com as suas diferentes capacidades. As abordagens utilizadas pela indústria de comunicação móvel num passado recente, envolviam normalmente a gestão do compromisso entre a qualidade do serviço prestado e a abrangência dos dispositivos do mercado alvo. Apesar de esta abordagem resultar na maioria dos casos, a sua adopção implica problemas de escalabilidade dos serviços além de um aumento considerável na manutenção e actualização dos mesmos devido à sua rápida e constante evolução na vertente tecnológica. Este facto reveste-se ainda de maior importância quando o âmbito destes serviços engloba a utilização da Internet e da Web. Nesse sentido, a procura de uma solução que tenha em conta estas considerações revela-se proveitosa na correcta gestão das aplicações acedidas por dispositivos móveis. A heterogeneidade atrás referida é derivada das duas componentes transversais a todos os dispositivos móveis: Hardware e Software. Sendo que os principais factores que afectam a sua descrição são:

- As capacidades ou características dos dispositivos. (O processamento da informação disponibilizada ao terminal deve ter em conta as suas capacidades no que concerne capacidade de processamento e memória, interfaces disponíveis, segurança da transmissão de informação, sistema operativo, formatos suportados, etc.).

- As linguagens de markup. (A entrega de conteúdo ao terminal deve ter em conta a sua linguagem de markup já que estas variam consoante o tipo de dispositivo, em que a maioria destes suporta apenas um subconjunto das referidas linguagens).

No capítulo Estado da Arte são descritas as ferramentas disponíveis para este propósito, o seu modo de funcionamento e principais vantagens oferecidas.

A modelização de uma arquitectura que adopte estes factores como requisitos para o desenvolvimento de aplicações móveis é apresentada no capítulo Arquitectura. Neste, é proposto um sistema de identificação do dispositivo juntamente com a extracção de modo automático das suas características. Além disso, é adicionado o suporte para as diferentes linguagens de markup com vista a adaptação de conteúdo disponibilizado por serviços Web. A implementação do sistema e os resultados obtidos são analisados nos capítulos seguintes. O conceito utilizado para resolver o paradigma da heterogeneidade dos dispositivos é discutido no capítulo Conclusões e são também tratadas as perspectivas de desenvolvimento futuro.

1.1 - Objectivos da Dissertação

O objectivo deste trabalho consiste no desenvolvimento de um módulo software para terminais móveis (PDAs e/ou telefones móveis) que execute as funções de um agente de contexto. Um agente de contexto é um módulo que consegue extrair de uma forma automática características sobre recursos hardware e software do terminal em que reside e de sinais gerados por componentes ou periféricos desse terminal. Recursos hardware incluem o tipo de interface de rede disponível, o tipo de processador, existência e tipo de câmara vídeo, microfones, capacidade de memória, etc. Recursos software incluem tipos de codecs multimédia instalados, tipo e versão do browser Web, aplicações software instaladas no terminal, etc. Sinais gerados por componentes ou periféricos do terminal podem incluir sinais de vídeo captados por uma câmara ou sinais de áudio captados por um microfone, etc.

Esta informação é designada de informação de contexto e pode ser utilizada para auxiliar a tomada de decisão sobre a necessidade de adaptar conteúdos que são enviados para o terminal de modo a satisfazer limitações ou características do terminal e do contexto de utilização (como por exemplo condições ambientais - luminosidade extrema, ruído ambiental elevado, etc).

1.2 - Definição de Contexto

“Contexto é tudo o que nos rodeia”. A partir desta expressão tenta-se definir o termo que na Ciência de Computação Móvel é usado para descrever o meio físico que circunda o uso e operação de um dispositivo móvel.

O termo “contexto” aplicado à Computação Móvel é por vezes difuso e sujeito a interpretações diferentes. Numa tentativa de categorização do conceito de contexto são propostas duas abordagens ao termo:

- categorização operacional
- categorização conceptual

A primeira está intrinsecamente relacionada com o modo de como a informação de contexto é obtida, modelizada e processada. Em analogia poder-se-á identificá-la como informação de contexto de baixo nível.

A segunda engloba as características dinâmicas e de fiabilidade que a informação de contexto pode sofrer, estando mais relacionada com o ambiente e cenários de utilização. Analogamente poderemos identificá-la como informação de contexto de alto nível.

A maioria das definições do termo contexto na literatura tende a recair sobre uma das duas anteriores categorizações. No presente trabalho será considerada principalmente a primeira.

Alguns autores preferem usar o termo “contexto físico” precisamente para distingui-lo de outros tipos de contexto presentes na Computação Móvel cujo âmbito de investigação reflecte a natureza de um estado de uma entidade para além da captura das condições físicas em seu redor. Tais estados poderão ser sistemas circundantes ou análise de infra-estruturas de rede presentes. Esse esforço de caracterização e modelização é por vezes englobado no uso do termo “contexto-situacional”.

Quando esse termo é empregue, normalmente é inferida uma análise prévia substancial e fusão de dados adquiridos por sensorização individual específica capazes de descrever vários tipos de cenários de utilização, comportamentos do utilizador, técnicas de interactividade, etc.

1.3 - Context-Aware

Este tipo de informação, a um nível mais geral, pode ser considerada como informação que é utilizada para caracterizar uma situação/estado de uma entidade. A entidade pode ser uma pessoa, um lugar ou um objecto considerado relevante. A informação contextual é obtida pela observação das entidades relevantes através de sensorização específica. Cada sensor de contexto é responsável pela aquisição de informação contextual e além disso deve garantir o encapsulamento do modo como a informação é obtida, das entidades superiores.

Os dispositivos em que a análise multi-sensorial e não menos importante, o seu processamento, é tida em conta no fornecimento de informação ou serviços ao utilizador em que a relevância dessa informação é dependente da acção/comportamento deste, são habitualmente designados de “context-aware”.

Este processamento pode ser efectuado pela infra-estrutura e posteriormente reportado ao dispositivo (indirect context-awareness) ou efectuado na sua totalidade pelo dispositivo (direct context-awareness).

A escolha do tipo de classificação de contexto deve então ser considerada tendo em conta o tipo de informação que se pretende recolher e a sua relevância para o papel do agente de contexto. Normalmente, o contexto aplicado aos dispositivos móveis recai na primeira classificação dados os seus poucos recursos.

Devido à dinâmica inerente a este tipo de operação, limitação dos dispositivos, fontes distribuídas, problemas de continuidade, etc., a probabilidade da informação adquirida não ser perfeita, é real.

Para lidar com este problema alguns sistemas tentam caracterizar a veracidade da informação de contexto obtida. Tal caracterização deve ser implícita na modelização do sistema que capta a informação e em grosso modo pode ser classificada em quatro tipos: desconhecida, ambígua, imprecisa e errada.

Adicionando um predicado de probabilidade a cada um dos tipos de informação e mostrando essa indicação ao utilizador no momento da tomada de decisões, esta acção trará comprovadamente uma melhoria no número de decisões certas ao longo do tempo.

Paralelamente à incerteza associada à informação, a tentativa de previsão de um estado de contexto é ainda mais duvidosa ou contingente.

Por outro lado, podemos tentar inferir tal estado recorrendo à análise de padrões comportamentais, os quais requerem o registo e armazenamento de dados, porém.

1.4 - O agente

Um agente pode ser definido da seguinte forma:

Um agente é uma instância de um objecto de software que está situado num ambiente de execução e possui as seguintes propriedades obrigatórias:

- *Reacção* - monitoriza mudanças no ambiente e reage de acordo com essas mudanças;
- *Autonomia* - dispõe de controlo sobre as suas próprias acções;
- *Orientado ao objectivo* - é pro-activo;
- *Temporalmente contínuo* - executa continuamente;

e pode possuir uma ou mais das seguintes propriedades ortogonais:

- *Comunicativo* - pode comunicar com outros agentes;
- *Móvel* - pode viajar de um host para outro;
 - *Aprendizagem* - adapta-se de acordo com eventos/experiência anterior;
- *Credível* - apresenta credibilidade ao utilizador final.

A mobilidade é uma propriedade ortogonal dos agentes, na medida em que nem todos os agentes necessitam de serem móveis. Um agente pode simplesmente manter-se estático e comunicar com o meio circundante por meios convencionais.

Estes meios incluem várias formas de comunicação remota por procedimentos de estabelecimento de ligações ou por simples notificação.

Os agentes que não possuem mobilidade são denominados agentes estacionários. Por oposição, um agente móvel não está confinado ao sistema onde iniciou a execução, é livre de viajar entre os diversos hosts ligados à rede. Uma vez instanciado num ambiente de

execução, consegue transportar o seu “estado” e “código” para outro ambiente de execução onde retoma a sua actividade.

O conceito de “estado” do agente reflecte o conjunto de valores de determinados atributos que ajudam o agente a determinar a próxima acção a desenvolver quando este resumir a execução no seu destino. O termo “código” é aqui empregue, supondo um ambiente de contexto orientado ao objecto, para referenciar as classes necessárias para o agente executar.

Apresenta-se em seguida algumas vantagens decorrentes do uso de sistemas com agentes:

1. - Redução da carga da rede.
2. - Contorno da latência da rede.
3. - Encapsulamento de protocolos.
4. - Execução assíncrona e autónoma.
5. - Adaptação dinâmica.
6. - Natureza heterogénea.
7. - Robustez e tolerância a falhas.

O advento da tecnologia baseada em agentes traz inúmeras vantagens, na medida em que providencia serviços em que o acesso e manipulação de informação são especificados e disponibilizados consoante certo tipo de propriedades tais como a localização, hora e ambiente que envolve o utilizador. Para o fazer o agente tem de ser capaz de reunir informação designada de “contexto”.

Na tentativa de migrar a tecnologia e conceito de agente de software adaptado ao contexto convém definir e caracterizar o papel do agente de contexto.

Um agente de contexto é basicamente um serviço de descoberta e captação de informação que representa uma entidade e mantém referências a determinadas fontes de contexto que fornecem informação sobre essa mesma entidade. As entidades consideradas podem ser as mais variadas e possuir propriedades diversificadas.

Dada uma referência ao agente de contexto, uma aplicação pendente pode questionar o agente sobre determinado tipo de informação de contexto, à qual o agente responde com uma referência a uma fonte de contexto próxima que se situa vulgarmente entre a aplicação e a fonte de contexto legítima para fornecimento dessa informação.

A inclusão de tal fonte de contexto “proxy” visa reforçar a política de privacidade definida pelo utilizador do agente de contexto e permite também reduzir a quantidade de recursos que os nós móveis da rede necessitam de suportar para hospedar fontes de contexto.

Partindo deste ponto, é necessário ter em atenção as restrições associadas à maioria dos dispositivos móveis tais como a capacidade de memória disponível, o poder de processamento, a diversidade de interfaces e tamanho do visor.

Desse modo, o agente tem de ser capaz de captar e filtrar informação relevante para as entidades fornecedoras de serviços e deve também assegurar o funcionamento e correcta operação e transferência de informação sem a intervenção do utilizador.

Nesse sentido, o perfil do agente de contexto móvel adequa-se completamente.

Capítulo 2

Estado da Arte

A comunicação e transferência de informação de contexto entre o agente do terminal móvel e o fornecedor de serviços pressupõem uma normalização da informação de contexto, sem a qual se torna inviável a posterior comunicação e entrega de conteúdo multimédia. Este processo envolve a modelização de uma arquitectura que permita o reconhecimento de informação em tempo real e na presença de condições adversas ou de variação rápida, com vista a disponibilizar informação às aplicações baseando-se numa topologia assente em geração de eventos. Desse modo, torna-se possível a descrição de mudanças de ambiente de utilização para posterior interpretação pelo agente de contexto.

Um dado importante a reter no desenho de tal arquitectura é explicitar bem o objectivo e serviço para a qual esta foi criada tendo sempre presente o tipo de aplicações que esta suporta.

Apesar de parecer uma frase redundante, é necessário frisar este ponto já que a maioria dos trabalhos desenvolvidos nesta área (processamento de informação de contexto/adaptação de conteúdos para dispositivos móveis) diferem precisamente na modelização da sua estrutura consoante os parâmetros necessários à adaptação das aplicações entregues ao utilizador. Como a própria adopção do uso de contexto subentende, a finalidade é atingir um modo de operação completamente independente das características do dispositivo, mobilidade do utilizador, características da rede, etc.

2.1 Detecção de dispositivos

O avanço das tecnologias de comunicação móveis juntamente com a rápida adopção de dispositivos móveis com capacidade de aceder à Internet, enriqueceram a experiência de uso destes por parte dos utilizadores. No entanto, dada a heterogeneidade dos dispositivos e particularmente, das suas capacidades (tamanho do visor, capacidade de memória, capacidade de processamento, etc.) essa experiência rapidamente se torna limitada. Isto devido ao facto do conteúdo disponibilizado (documentos, imagens, áudio e vídeo) não ser servido no melhor formato ou mesmo não suportado devido às capacidades restritivas dos dispositivos. Adicionalmente, os dispositivos móveis em questão podem correr diferentes sistemas operativos (Symbian, Windows Mobile, Linux, Android) e podem suportar diferentes linguagens de markup, tais como WML, cHTML ou XHTML. Assim qualquer operação de adaptação de conteúdo requer a identificação do dispositivo em questão *a priori*.

O processo de detecção de dispositivos móveis convencional é geralmente descrito e aplicado no contexto da utilização de serviços web. Este processo requer que um utilizador visite uma página web através do seu dispositivo sendo a identificação desencadeada por este evento. Quando um dispositivo envia um pedido HTTP, o protocolo automaticamente adiciona informação nos cabeçalhos do pedido. Destes, os mais relevantes no processo de detecção do dispositivo são os seguintes: user-agent, x-wap-profile e accept. O processo de identificação do dispositivo utiliza a informação contida nestes cabeçalhos, porém nem todos os dispositivos cumprem as recomendações dos standards em conformidade. Ocasionalmente os cabeçalhos contêm informação falsa ou mesmo a ausência desta. Outras vezes, essa informação é proprietária ou é alterada por servidores proxy em alguns nós da rede. Estas incoerências introduzem obstáculos ao processo de detecção caracterizando-o num problema em constante mutação.

Foram já propostos alguns modelos de adaptação de conteúdo que diferem precisamente no local onde o processo de adaptação é efectuado. Num esforço de categorização deste último, o órgão W3C categorizou três tipos de adaptação de conteúdo descrito em seguida.

A detecção do dispositivo pode ser efectuada em várias etapas no processo da entrega do conteúdo. As mais vulgares são detecção do lado do cliente, detecção por um proxy de rede e detecção do lado do servidor.

A detecção do lado do cliente pressupõe a execução de algum tipo de script no dispositivo antes da entrega do conteúdo ao utilizador. Um caso frequente de utilização é execução de um javascript que redirecciona o dispositivo móvel para outro web-site com conteúdo especificado para o efeito. No entanto, esta abordagem é propícia a erros devido à diferenciação no modo como diferentes dispositivos tratam os mesmo scripts e o resultado pode não ser o esperado. Por outro lado, podem surgir eventuais limitações no processo devido ao fraco poder de processamento ou à reduzida largura de banda do dispositivo em questão.

A detecção por via de um proxy de rede é feita num nó intermediário da rede e não permite ao proprietário do conteúdo controlar o processo de detecção salvo se o proxy pertencer ao sistema que disponibiliza o conteúdo. Isto porque o proxy poderá modificar a informação dos cabeçalhos, redimensionar imagens, modificar a estrutura do web site ou até reescrever o URL pedido sem o conhecimento do proprietário do conteúdo que tenta efectuar o processo de detecção. No pior cenário, é possível ao proxy modificar o pedido enviado ao servidor do conteúdo e originar um pedido para um dispositivo diferente deteriorando desta forma o processo de detecção.

A detecção do lado do servidor que disponibiliza o conteúdo a oferecer possibilita ao proprietário escolher o modo como diferentes dispositivos são tratados. Este tipo de detecção juntamente com algum tipo de repositório de capacidades de dispositivos permite otimizar o conteúdo oferecido ao dispositivo requerente baseando-se nas suas capacidades.

O propósito final da detecção do dispositivo é a obtenção das características estáticas do terminal que requisita o conteúdo. Essa informação reveste-se de grande importância na eventual necessidade de adaptação de conteúdo multimédia a ser entregue visto que a modelização de um sistema dessa natureza tem como requisito obrigatório o conhecimento prévio das capacidades do dispositivo do utilizador.

2.2 Adaptação de conteúdo

A execução de aplicações em diferentes dispositivos ou num ambiente de computação móvel solicita aquisição de dados e promove a interacção entre o utilizador final e os serviços a este disponibilizados. Contudo, tal abordagem requer muitas vezes a adaptação de conteúdo.

A adaptação pode ser definida como a capacidade de adequação de uma aplicação a mudanças nas circunstâncias do ambiente em que executa. Tal adequação pode ocorrer com ou sem a intervenção humana. Essa característica pode introduzir barreiras no desenvolvimento de aplicações adaptativas ao contexto.

Dentro desse contexto, a adaptação de uma aplicação pode ser caracterizada de três modos:

- **Adaptação à descrição:** é a capacidade de descrever uma aplicação permitindo a sua adequação ou transformação em diferentes linguagens ou plataformas de programação. Nesse caso, a aplicação é descrita numa meta-linguagem e é transformada numa linguagem de programação (e.g. Java, C++) ou em uma linguagem de markup de hipertexto (XHTML, cHTML, WML).

- **Adaptação ao dispositivo:** é a habilidade de uma aplicação em adequar o seu modo de execução às características do dispositivo. Essas características podem ser estáticas, como número de cores e dimensões do visor, e dinâmicas como a quantidade de memória e bateria disponíveis.

- **Adaptação ao contexto:** é a propriedade de uma aplicação em adequar-se a mudanças no contexto em que executa. As mudanças no contexto podem ser decorrentes das alterações da localização do dispositivo, do interesse do utilizador e da largura de banda do canal de comunicação.

Estes três tipos de adaptação não estão restritos a uma mudança no modo de execução das aplicações nos dispositivos móveis. A adaptação pode estar relacionada também à adequação dos dados que a aplicação acede. Obviamente que o mesmo conteúdo direccionado para um *notebook* não deve ser enviado a um celular, por exemplo. Para isso, deve haver tanto a redução das informações, como a modificação do formato da apresentação dos dados. Entretanto, essa adaptação de conteúdo não se baseia somente nas características dos dispositivos. As preferências e o contexto do utilizador que utiliza a aplicação também podem ser relevantes. Algumas informações sobre a localização, o perfil e os parâmetros de qualidade de serviço do usuário são importantes para a decisão relacionada à filtragem da

informação. Essa adaptação é chamada de adaptação semântica, podendo levar em consideração, inclusive, o idioma do utilizador.

O desenvolvimento de aplicações para dispositivos móveis é difícil devido ao simples facto de diferentes dispositivos apresentarem capacidades diferentes. Este aspecto é muitas vezes chamado de fragmentação do mercado de dispositivos. Esta diversidade está presente em múltiplas dimensões:

- Características Hardware (tamanho do visor, teclado numérico ou QWERTY, etc.)
- Características da rede (velocidade da ligação, largura de banda)
- Formatos Multimédia (GIF, JPEG, PNG, WBMP, AMR, MIDI, MPEG3, MPEG4, 3GPP, etc.)
- Comportamento do browser (formatação, suporte da linguagem de markup varia bastante, layout das imagens, etc.)

Esta diversidade nasceu de vários factores:

- Fabricantes dos dispositivos apostam no factor de diferenciação entre eles.
- Operadores de rede tentam vender dispositivos que são únicos ou altamente adaptados às suas redes.
- Crescente inovação tecnológica e cumprimento de normas reguladoras
- Software residente nos terminais não é actualizado com muita frequência.

A consequência resultante destes pontos é a preservação da fragmentação do mercado de dispositivos.

O surgimento da adaptação de conteúdo reveste-se de gradual importância neste contexto sendo ainda mais relevante quando o modelo de desenvolvimento de aplicações é o da Web. A aprendizagem das capacidades dos dispositivos é um factor crucial para esse fim. Algumas aplicações móveis conseguem captar características do terminal o que permite o envio de conteúdo previamente modificado. Este processo em que o conteúdo é transformado é apelidado de adaptação de conteúdo ou *Multiserving*. Convém referir que o termo adaptação de conteúdo é utilizado por alguns autores para referenciar servidores proxy que modificam e manipulam o conteúdo Web ao longo da rede para servirem os clientes. Não obstante, esse comportamento não é o desejável para o desenvolvimento da aplicação deste trabalho já que o conteúdo deve ser gerado especificamente de acordo com as características do terminal.

A adaptação de conteúdo pode ser conseguida de várias formas:

- Redireccionamento dos dispositivos para um site com uma versão adequada a este (XHTML ou WML).
- Adicionar ou esconder links para conteúdo que pode ser descarregado consoante o suporte disponibilizado pelo dispositivo.
- Envio dos tipos MIME e codificação correcta dos caracteres.

Normalmente o processo de adaptação requer a inspecção dos cabeçalhos http enviados por um dispositivo para prever as funcionalidades suportadas. Seguindo esta abordagem é possível uma adaptação dinâmica de conteúdo e a validação correcta da linguagem de markup específica dos dispositivos.

Esta validação é crucial no contexto da Internet móvel por causa das diferentes necessidades de convergência entre as linguagens de markup existentes. A grande fatia é englobada no XHTML MP que foi criado para convergir com o HTML, o suporte da Internet. Por outro lado, as implementações do XHTML MP diferem de browser para browser, podendo originar quebras de execução das aplicações devido a pequenas diferenças nem sempre perceptíveis das versões do XHTML MP quando utilizado para suportar todos os dispositivos WAP 2.0. Aliado a esse facto, os dispositivos mais antigos usam o WML que não é compatível na totalidade com o XHTML MP. O panorama não melhora se tomarmos em consideração outra variante de markup existente em algumas regiões GSM da Ásia, o cHTML (compact HTML) que dá suporte aos dispositivos que operam em i-mode.

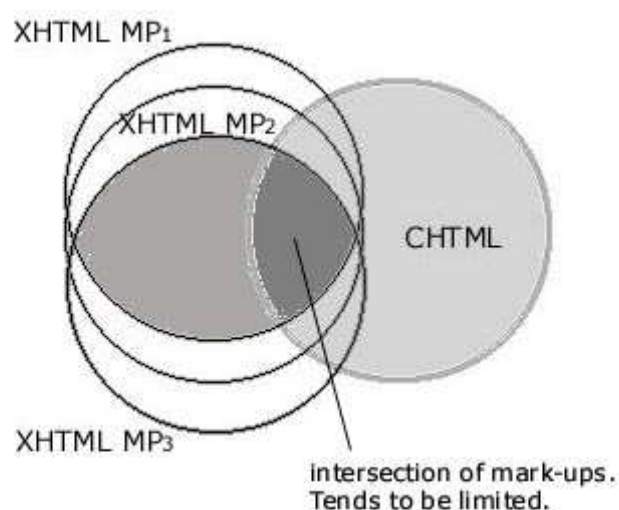


Figura 1. - Intersecção das linguagens de markup de dispositivos móveis.

Ao identificar a estrutura comum entre as diferentes linguagens de markup tornou-se possível gerar os elementos constituintes de cada linguagem (tags) dinamicamente. Este processo cria uma camada abstracta que opera por cima do conteúdo original. Seguidamente, procede ao envio deste, tal como está, aos dispositivos mais capazes ou degrada-o subtilmente de forma a ser compatível com os dispositivos de gerações anteriores.

Depois de ultrapassado o problema de apresentação da informação a ser enviada aos dispositivos móveis e para obter maior eficiência na tomada de decisão da necessidade de adaptação de conteúdo a ser transmitido, convém explicitar os requisitos para efectuar tal decisão. Essa decisão adquire assim o carácter de contexto propriamente dito.

Requisitos de um sistema de aquisição de contexto:

- 1) Identificação do dispositivo do cliente.
- 2) Utilização e gestão de perfis de utilizadores.
- 3) Adaptação de conteúdo adequado ao dispositivo do cliente.
- 4) Utilização de cache.

Um sistema de tais características é normalmente implementado por uma de duas abordagens: adaptação de conteúdo baseada em agentes móveis e adaptação de conteúdo baseada em *proxies* ou *gateways*.

Na primeira, a aplicação no dispositivo móvel contém um agente móvel capaz de extrair dados sobre o dispositivo e o utilizador migrando posteriormente para uma rede com o intuito de adquirir informações para a aplicação.

Na segunda abordagem, imagens, vídeos, áudio e páginas Web, são requisitadas pelas aplicações através de *gateways* ou *proxies* que são responsáveis por adaptar o conteúdo utilizando as características do dispositivo ou do utilizador para esse fim.

A informação designada de contexto estática pode ser adquirida através do repositório de dispositivos implementado neste sistema após a correcta identificação do dispositivo via user-agent.

Porém, a informação designada de contexto dinâmico reveste-se de grande importância para cumprir o objectivo descrito no parágrafo anterior. Esse tipo de informação é normalmente identificado como as quantidades de memória de execução e de memória de armazenamento livre, o tipo de imagem e vídeo suportado pela câmara do dispositivo e a quantidade de bateria disponível. Esta informação dinâmica pode influenciar a execução de uma aplicação

que esteja a operar no dispositivo e poderá fornecer indicações cruciais sobre o modo de operação do dispositivo em tempo real e a consequente estruturação da adaptação de conteúdo a enviar.

Para se construir aplicações adaptativas, algumas questões devem ser respondidas:

- (i) O modo de obtenção dos perfis dos dispositivos e dos utilizadores.
- (ii) Quais as políticas de adaptação empregues
- (iii) Quando deve ser realizada a adaptação.

Estas questões podem ser respondidas pela inclusão de uma entidade, o servidor *proxy*. Um servidor *proxy* adapta os recursos a cada requisição e é responsável por delegar como e quando se deve realizar a adaptação.

No entanto, com vista a cumprir o objectivo da dissertação deverá ser proposto a criação de um módulo que execute no dispositivo e que possua algumas das propriedades de um agente.

Esse módulo seria responsável por monitorizar eventos decorrentes da variação das capacidades dinâmicas do dispositivo, tais como a capacidade disponível da bateria, a memória de execução e de armazenamento livre e a capacidade de incluir sinais de vídeo ou áudio periféricos.

Esse tipo de sinais pode ser descrito programaticamente pela inclusão da API Java MMAPI (JSR-135) que possibilita a captura de áudio e vídeo pelo dispositivo e a partilha dessa informação na rede em que este opera. Desse modo, o cruzamento de informação estática e dinâmica das capacidades do dispositivo traria grandes vantagens na necessidade de adaptação de conteúdo.

Adicionando essa informação ao registo das preferências e do comportamento de cada utilizador através da criação de perfis de utilizadores no servidor, é possível criar um sistema mais sensível ao contexto-situacional ou de alto nível.

2.3 WURFL

O objectivo do projecto de WURFL é descrever as capacidades da generalidade dos dispositivos móveis em torno do planeta e fornecer uma API simples para questionar o repositório acerca de determinada capacidade. A necessidade de especificar as capacidades de um dado dispositivo móvel com vista a desenvolver aplicações para estes tornou evidente a procura de soluções eficazes por parte da indústria. Em resposta a esta necessidade surgiu o WURFL como o esforço conjunto da comunidade de desenvolvimento web open source.

Tendo em conta que a manutenção de uma base de dados actualizada com estas características é uma tarefa morosa, foi decidido pelos criadores do projecto expandi-lo a todos os colaboradores interessados no desenvolvimento deste software.

Dada a heterogeneidade de dispositivos existentes e a natureza abrangente da própria Internet, o WURFL tem como premissa a disponibilidade na maioria das plataformas de desenvolvimento actuais. Desse modo, o formato escolhido para descrever as capacidades dos dispositivos foi o XML (actualmente a maioria da linguagens de programação disponibilizam API's que suportam XML).

Considerando que uma base de dados local contendo informação sobre todos os dispositivos móveis identificáveis seguramente resultaria num ficheiro de grande dimensão (à volta de 400 dispositivos e 40000 capacidades) e acrescentado o facto de cada dispositivo poder ter diferentes subversões (muitas vezes correspondendo a uma diferente versão de firmware), será legítimo depreender que a matriz de capacidades rapidamente tornará o seu processamento moroso e revela tendência a aumentar com o tempo. A sua manutenção e não menos importante a sua actualização revela-se uma tarefa difícil.

No entanto, existem algumas considerações que podem ajudar o desenvolvimento de uma solução compacta, sendo estas:

- browsers de diferentes fabricantes são intrinsecamente não coincidentes mas também possuem várias características em comum entre eles.
- Dispositivos/browsers do mesmo fabricante são frequentemente uma evolução do mesmo hardware/software.

- Dispositivos de diferentes fabricantes podem correr o mesmo software. Ex. O browser de muitos dispositivos das marcas Siemens, Motorola, Alcatel, Mitsubishi, Samsung, Panasonic é fornecido pela Openwave Systems.

A eficiência do WURFL está relacionada com a exploração destes factos por via programática. Baseando-se no conceito de famílias de dispositivos em que a raiz do sistema é o ‘generic device’, o WURFL torna-se simples de actualizar bastando para isso gravar no sistema o ficheiro patch mais recente que é também disponibilizado pelo projecto.

O WURFL é então um ficheiro único XML com a seguinte estrutura:

```
<wurfl>
  <devices>
    <device ...>
      <group ...>
        <capability name="..." value="...">
          :
        </group>
        :
      </device>
      :
    </devices>
  </wurfl>
```

Esta extensa lista de dispositivos, em que cada um destes é uma lista de capacidades forma o ficheiro wurfl.xml. Apesar disso, um pedido de identificação de um dispositivo não obriga ao processamento integral da matriz (dispositivosXcapacidades) já que em média um dispositivo novo no mercado tem cerca de 90% de probabilidade do seu software ser da mesma versão de um dispositivo anterior do mesmo fabricante. Este facto é explorado pelo WURFL quer na interrogação do repositório, quer na inserção de novos dispositivos. O simples conhecimento da User-Agent string do dispositivo é suficiente para instruir o repositório que as capacidades do dispositivo devem ser inferidas de um dispositivo já existente. Este mecanismo foi apelidado de “fall-back” e mostra-se extremamente útil na descrição de diferentes sub-versões de software do mesmo dispositivo e também na tarefa de manter o tamanho total do ficheiro logaritmicamente compacto em relação à quantidade de dados nele contidos.

```

<device user_agent="Nokia3100" actual_device_root="true"
  fall_back="nokia_generic_series40"
  id="nokia_3100_ver1">
  <group id="product_info">
    <capability name="model_name" value="3100"/>
  </group>
  <group id="xhtml_ui">
    <capability name="xhtml_format_as_css_property" value="true"/>
    <capability name="xhtml_supports_table_for_layout" value="true"/>
    <capability name="xhtml_supports_css_cell_table_coloring" value="true"/>
    <capability name="xhtml_readable_background_color1" value="#99CCFF"/>
    <capability name="xhtml_readable_background_color2" value="#FFFFFF"/>
  </group>
  <group id="markup">
    <capability name="preferred_markup" value="html_wi_oma_xhtmlmp_1_0"/>
    <capability name="html_wi_w3_xhtmlbasic" value="true"/>
    <capability name="html_wi_oma_xhtmlmp_1_0" value="true"/>
  </group>
</device>

```

A tag <device> descreve um dispositivo e contém no mínimo as seguintes capacidades:

- A string do “User-Agent”

- O “device ID” (identificador de um dispositivo dentro do WURFL)

- O atributo “fall_back” que permite inferir mais capacidades do dispositivo (o valor deste atributo é o “device ID” de outro dispositivo).

Além destes dados, um <device> pode ainda conter informação sobre as características do terminal móvel denominadas de <capabilities> e o atributo actual_device_root="true" usado para identificar dispositivos reais.

As tags <group> são usadas somente para melhorar a leitura do código e para agrupar capacidades semelhantes. A tag <capability> contém informação sobre uma característica do dispositivo e todos os dispositivos referenciados por “fall-back”. Assim, seguindo a cadeia de

“fall-back” é possível navegar pela família de dispositivos da mesma versão. No entanto, no fim da cadeia reside o dispositivo “generic” que é a abstração do dispositivo mais geral possível e cujo propósito é garantir que as queries para um valor de uma capacidade retornem sempre um resultado.

Para inquirir o ficheiro XML são disponibilizadas várias API's, sendo as mais importantes as de Java e PHP. Estão também disponíveis API's para Perl, Ruby, DotNet e Python. Independentemente da API escolhida para implementar o WURFL, existem dois métodos principais transversais às demais:

- Dada uma User-Agent string retorna um WURFL device id.
- Dados um WURFL device id e um nome de uma capacidade, retorna o valor dessa capacidade.

Combinando estes métodos facilmente se obtém os valores de até 300 capacidades para um dispositivo específico através do processamento do pedido HTTP original.

A classe UAManager dispõe de dois métodos diferentes para realizar esta tarefa:

- getDeviceIDFromUA(String user-agent)

retorna a String que representa um WURFL device ID. Esta função procura uma correspondência exacta no WURFL. Em caso negativo, será retornado o device “generic” (i.e. o ID do dispositivo não reconhecido).

- getDeviceIDFromUALoose (String user-agent)

também retorna a string que representa um WURFL device ID. A diferença em relação ao método anterior reside na persistência em obter, no mínimo, uma correspondência semelhante. Em caso de falha na obtenção de uma correspondência exacta são aplicadas as seguintes heurísticas para encontrar um dispositivo razoavelmente semelhante:

1. O processamento da string do User-Agent é iniciado com a remoção dos Chars do fim da string, um de cada vez. Em cada iteração é feita uma tentativa de correspondência entre a string e a parte inicial de cada user-agent existente. O ciclo termina em caso de sucesso na correspondência ou em caso da string processada ter ficado reduzida a cinco Chars.
2. O operador móvel Vodafone introduziu na sua rede diversos dispositivos que adicionam a string “Vodafone/” no início de cada user-agent, o que poderia

reduzir a taxa de correspondências positivas. Esses dispositivos foram devidamente listados no WURFL e em caso de falha de correspondência a string “Vodafone/” é removida e o algoritmo de correspondência é aplicado do início.

3. Se não for encontrada nenhuma correspondência, o algoritmo de correspondência procura por pistas que possam indicar algum pressuposto, tais como:
“UP.Browser/4”, “UP.Browser/5”, “UP.Browser/6”, “Series60”, “Mozilla/4.0”, etc.
4. Em caso de ausência total de correspondências é retornado o device “generic”.

Aliada à persistência do algoritmo de correspondência existe o facto de as pesquisas serem guardadas em cache, o que permite que pesquisas com dados anteriores sejam resolvidas com a simples procura num HashMap.

Obtenção dos Valores de Capacidades

Depois de encontrar um device ID a pesquisa das suas capacidades é trivial, bastando para isso usar o método `getCapabilityForDevice()` para obter o valor de qualquer capacidade. Este método pertence à classe `CapabilityMatrix`.

- `getCapabilityForDevice(String deviceId, String capabilityName)`

Retorna o valor de uma capacidade para um dispositivo específico definidos no argumento do método

Exemplo (Java API):

```
System.out.println(cm.getCapabilityForDevice("sie_m50_ver1", "resolution_width"));
```

A API Java do WURFL:

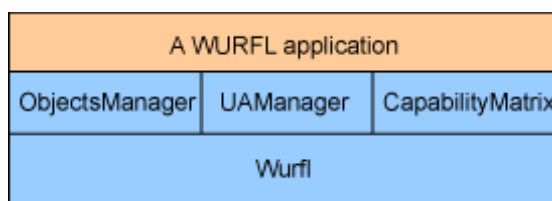


Figura 2. Classes da API Java do WURFL implementadas numa aplicação.

A API Java do WURFL contém quatro classes: `Wurfl`, `ObjectsManager`, `UAManager` e `CapabilityMatrix`. Estas classes são usadas no processo de aprendizagem das capacidades do dispositivo. As classes `ObjectsManager`, `UAManager` e `CapabilityMatrix`

situadas entre a camada da aplicação e a camada da classe Wurfl, usam por turnos os métodos desta última que também inclui o ficheiro wurfl.xml.

A classe WURFL tem vários construtores, todos eles protegidos de modo a que não seja possível instanciar a classe WURFL directamente. É recomendado utilizar a classe ObjectsManager para instanciar um objecto WURFL.

O primeiro construtor tem um único argumento que é o caminho no sistema de ficheiros local para o ficheiro wurfl.xml. Analisa o conteúdo do ficheiro e fica habilitado para servir pedidos para capacidades específicas de dispositivos.

O segundo construtor tem dois argumentos do tipo string. O primeiro argumento especifica o caminho no sistema de ficheiros local para o ficheiro wurfl.xml. O segundo argumento especifica o caminho no sistema de ficheiros local para o ficheiro chamado wurfl_patch. O WURFL usa o ficheiro patch para permitir às aplicações actualizarem as características das capacidades dos dispositivos. Este facto reveste-se de grande importância devido ao constante aparecimento de novos dispositivos com capacidades melhoradas.

O terceiro construtor efectua a mesma função do terceiro mas recebe como argumentos as referências aos objectos dos ficheiros wurfl.xml e wurfl_patch.xml em vez de strings com o caminho para estes.

Estes três construtores cumprem a mesma função que é a abertura do ficheiro wurfl.xml e do ficheiro wurfl_patch.xml se disponível, analisam o seu conteúdo (dispositivos, caminho de fall-back, grupos e capacidades), carregam os seus valores para variáveis internas e preparam o objecto WURFL para disponibilizar as capacidades de dispositivos.

Os métodos mais importantes da classe WURFL que também são usados por outras classes são os seguintes:

- isCapabilityDefinedInDevice()
- getCapabilityValueForDeviceAndCapability()
- getDeviceIDFromUA()
- getDeviceIDFromUALoose()

O método isCapabilityDefinedInDevice() aceita dois argumentos do tipo string para verificar se uma capacidade está definida para um determinado dispositivo. O método retorna true se a capacidade estiver definida e false em caso contrário.

O método getCapabilityValueForDeviceAndCapability() aceita os mesmos dois parâmetros anteriores e retorna o valor da capacidade pretendida. Este método analisa a totalidade do caminho de fall-back para encontrar o valor da capacidade.

O método `getDeviceIDFromUA()` aceita como argumento a string do user-agent do browser do dispositivo e retorna o identificador correspondente ao dispositivo desse user-agent dentro do WURFL. A string do user-agent deve ser extraída do cabeçalho user-agent do pedido http original.

O método `getDeviceIDFromUALoose()` cumpre a mesma função mas aceita como argumento o pedido http completo e retorna o identificador do dispositivo que originou o pedido http.

A classe `ObjectsManager` engloba funções da classe `WURFL` e dispõe de diversos métodos estáticos. O método `getWurflInstance()` que não aceita nenhum argumento retorna uma instância da classe `WURFL`. Para o fazer, o método precisa de saber a localização do ficheiro `wurfl.xml`. A localização deste último é obtida pela inspecção do ficheiro `wurfl.properties` no directório corrente.

Em alternativa, pode ser usado outro método denominado `getUAManagerInstance()` que instancia e retorna um objecto do tipo `UAManager`. Antes desta instanciação o método verifica se já existe uma instância do `WURFL`. Em caso negativo, instancia-o (como no método `getWurflInstance()`) e introduz o objecto `WURFL` no objecto `UAManager`.

Do mesmo modo, a classe `ObjectsManager` possui outro método chamado `getCapabilityMatrixInstance()` que retorna um objecto `CapabilityMatrix`. Este objecto também engloba um objecto `WURFL`.

Estão disponíveis também os métodos descritos anteriormente como `getDeviceIDFromUA()` e `getDeviceIDFromUALoose()` que oferecem as mesmas funcionalidades.

A grande vantagem de usar a classe `UAManager` em aplicações do lado do servidor reside na capacidade de memorizar as strings de user-agent e device id usadas em pesquisas anteriores. Quando a classe `UAManager` recebe um pedido de identificação de um dispositivo, verifica a sua cache para determinar se o device id está já disponível na memória. Se encontrar esse identificador, retorna-o sem consultar a classe `WURFL`. Assim, os métodos da classe `UAManager` executam com maior eficiência em comparação com os métodos da classe `WURFL`. Este facto é particularmente útil em aplicações do lado do servidor em que é expectável a ocorrência de pedidos repetidos para um mesmo dispositivo.

A classe `CapabilityMatrix` também dispõe de métodos análogos aos da classe `WURFL` (`isCapabilityDefinedInDevice()` e `isCapabilityDefinedInDevice()`) para descobrir as capacidades de um dispositivo. A única diferença reside na capacidade da classe `CapabilityMatrix` em manter pesquisas anteriores em cache, dotando-os de maior eficiência em relação aos da classe `WURFL`.

2.4 WALL

WALL é o acrónimo de Wireless Abstraction Library. É uma biblioteca de tags JSP que permite o desenvolvimento de aplicações Java para a Web com adaptação de conteúdo dinâmica envolvendo dispositivos móveis. Esta biblioteca tem como funcionalidade a entrega de conteúdo dentro de uma aplicação respeitando a linguagem de markup suportada pelo dispositivo. Tem a capacidade de interpretar essa linguagem internamente e devolver o conteúdo sob a forma de WML, XHTML-MP ou cHTML. Foi desenvolvida para ser usada em conjunto com a API's Java ou PHP do WURFL e está assente na seguinte arquitectura:

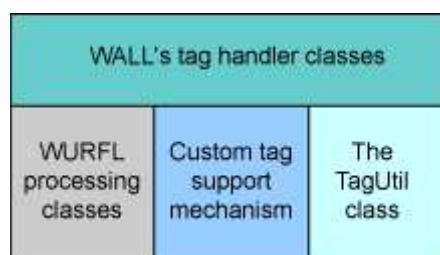


Figura 3. - Camadas de uma aplicação que utiliza a biblioteca WALL.

Para cumprir o seu propósito, a biblioteca utiliza internamente as classes de processamento do WURFL. Através deste mecanismo, descobre a linguagem de markup apropriada para o terminal móvel e transforma as tags JSP de uma aplicação Java para a Web em tags normalizadas nas três variações de markup dos dispositivos móveis.

Desse modo, é possível criar uma única versão de uma página JSP para ser acedida por diferentes dispositivos. Convém salientar que a adaptação dinâmica oferecida pela biblioteca refere-se a conteúdo relativamente simples e devidamente estruturado. Esse conteúdo engloba elementos comuns ao universo da Web assentes na linguagem html, tais como menus, listas de links, formulários simples, etc. Elementos mais complexos e de outros formatos como imagens, faixas de áudio ou vídeo não são suportados.

2.5 UAPROF

UAProf (User agent profile) é um documento XML que contém informação sobre o tipo do user-agent e capacidades de um dispositivo móvel. É um standart definido e mantido pela Open Mobile Alliance (anteriormente WAP Forum).

O UAProf foi criado tendo em conta o CC/PP (Composite Capabilities/Preferences Profiles) framework com o esquema RDF (Resource Description Framework). Tanto o CC/PP como o RDF foram definidos pelo W3C (World Wide Web Consortium).

Os perfis dos user-agent estão guardados num servidor designado de repositório de perfis. Habitualmente estes perfis são mantidos e actualizados pelos fabricantes de dispositivos móveis (i.e. os perfis de user-agent que descrevem as capacidades dos dispositivos móveis Nokia são armazenados num repositório mantido pela Nokia).

O URL que aponta para o perfil do user-agent de determinado dispositivo móvel está definido nos headers de um pedido HTTP enviado pelo dispositivo móvel. Muitas das vezes esse URL encontra-se no *x-wap-profile* header ou no *Profile* header. Por exemplo, o valor do *x-wap-profile* header gerado por um telemóvel Nokia 6230i é:

```
"http://nds1.nds.nokia.com/uaprof/N6230ir200.xml"
```

Um perfil de user-agent contém um determinado número de componentes e cada componente um determinado número de atributos. Efectuando o download do perfil de user-agent do Nokia 6230i é possível encontrar o atributo *ScreenSize* na componente *HardwarePlatform*:

```
...
<prf:component>
  <rdf:Description rdf:ID="HardwarePlatform">
    <rdf:type rdf:resource=
"http://www.openmobilealliance.org/tech/profiles/UAPROF/ccppschem-
20021212#HardwarePlatform"/>
  ...
    <prf:ScreenSize>208x208</prf:ScreenSize>
  ...
  </rdf:Description>
</prf:component>
...
```

Tal como mencionado anteriormente , o URL do documento UAProf dum dispositivo móvel está habitualmente localizado nos HTTP headers *x-wap-profile* ou *Profile*. Tomando como exemplo o Nokia 6230i, é possível extrair a seguinte linha:

x-wap-profile: "http://nds1.nds.nokia.com/uaprof/N6230ir200.xml"

No caso de impossibilidade do procedimento anterior, é ainda possível determinar o URL do documento UAProf seguindo os seguintes passos:

1. Verificar se existem headers *Opt* adicionais no mesmo pedido HTTP. Em caso afirmativo, ler o valor do *Opt* header que contenha o URL "http://www.w3.org/1999/06/24-CCPPexchange". O *Opt* header deve conter alguma informação semelhante a esta:

"http://www.w3.org/1999/06/24-CCPPexchange" ; ns=80

2. Extrair o número de namespace no *Opt* header. No exemplo anterior, o número do namespace é 80.

3. O URL do documento UAProf deverá estar localizado no header *NAMESPACE-Profile*, em que *NAMESPACE* é o número encontrado no *Opt* header. Exemplo:

80-Profile: "http://wap.sonyericsson.com/Uaprof/T68R502.xml"

Convém salientar que nem todos os dispositivos móveis suportam UAProf e como tal poderá não ser possível encontrar o URL do documento UAProf nos HTTP headers. Nesse caso, é ainda possível encontrar alguma informação sobre o user agent inspeccionando os HTTP headers tradicionais tais como *Accept*, *Accept-Charset*, *Accept-Language* e *User-Agent*.

Encontrado o URL de um perfil UAProf e após efectuado o seu download, para prosseguir com a extracção de valores associados a atributos do perfil é necessário seguir algumas regras já que a especificação UAProf é complexa comparando com a simples leitura de Headers HTTP *Accept* ou *User-Agent*. Destacam-se as seguintes regras da especificação:

- A especificação UAProf permite mais do que um único perfil para referenciar um dispositivo móvel o que significa que poder-se-á encontrar mais do que um URL no header *x-wap-profile*.

- Os perfis UAProf podem conter um bloco denominado default description block que é usado para atribuir valores predefinidos aos seus atributos. Estes valores podem estar especificados localmente no documento ou num perfil exterior.

- Para aplicar mudanças aos perfis de user-agent originais são usados dois headers específicos, a saber, *x-wap-profile-diff* e *Profile-diff*.

Para ler o valor final de um atributo é necessário seguir o processo chamado profile resolution:

- Leitura dos valores predefinidos dos atributos.
- Leitura dos valores dos atributos especificados nos perfis UAProf.
- Leitura das alterações especificadas nos headers *x-wap-profile-diff* ou *Profile-diff*.
- Combinar os valores obtidos anteriormente seguindo uma ordem e um conjunto de regras.

Quando é combinado um valor predefinido com um valor não predefinido, o valor predefinido é sempre substituído pelo valor não predefinido.

Quando um valor não predefinido é combinado com outro valor não predefinido, o parser do UAProf a regra de resolução do atributo. A especificação UAProf define três regras de resolução (Locked, Override e Append) que são usadas para instruir o parser do UAProf na combinação de valores não predefinidos. Todos os atributos estão associados com uma das três regras de resolução. O significado destas estão descritas abaixo:

Regra de Resolução	Significado
Locked	O valor final do atributo é o primeiro valor não predefinido. Ex: A regra de resolução do atributo <i>ScreenSize</i> é <i>Locked</i> .
Override	O valor final do atributo é o ultimo valor não predefinido. Ex: A regra de resolução do atributo <i>ColorCapable</i> é <i>Override</i> .
Append	O valor final do atributo é a lista de todos os valores não predefinidos. Ex: A regra de resolução do atributo <i>CcnpAccept</i> é <i>Append</i> . (O atributo

Actualmente, existem soluções open-source para interrogar o UAProf ou perfis CC/PP em forma de parser. A biblioteca DELI pode ser usada para esse fim desde que a implementação esteja em ambiente Java num Servlet.

Tal como mencionado anteriormente, a determinação de um valor final de um atributo envolve o seguimento de procedimentos específicos de acordo com a especificação UAProf. Contudo, os fabricantes de dispositivos móveis e os próprios operadores móveis preferem abordagens mais pragmáticas de parsing e na maioria das situações têm em consideração os seguintes factos:

- A existência de um único perfil UAProf para um dispositivo móvel que origina um único URL nos headers HTTP tais como *x-wap-profile*, *Profile*, etc.
- Preferencialmente, o perfil UAProf de um dispositivo móvel não engloba blocos descritivos predefinidos.
- Em caso de alterações adicionais ao perfil UAProf, ignorar os headers *x-wap-profile-diff* ou *Profile-diff* para esse efeito.

Deste modo, para conhecer o valor final de um atributo não necessário proceder às combinações de valores segundo as regras de resolução, bastando para isso a obtenção do URL do perfil do user agent. Após efectuado o download, é possível usar um método à escolha para capturar os valores dos atributos como por exemplo, simples correspondência textual ou Parsing em XML. A vantagem desta abordagem é a sua simplicidade e resulta a maior parte das vezes. A desvantagem é a eventual incorrecção dos dados adquiridos no casos dos três pontos anteriores não serem verdadeiros.

Componentes e Atributos UAProf

A especificação UAProf define um conjunto de componentes e atributos base de forma a que diferentes fabricantes de dispositivos móveis não cometam o erro de definir componentes e atributos diferentes para descrever a mesma capacidade. Por exemplo, o tamanho do visor de um telemóvel está sempre definido no atributo *ScreenSize* da componente *HardwarePlatform* independentemente do fabricante ser Nokia, Motorola ou Sony Ericsson. A lista completa dos componentes e atributos base está disponível no documento “UAProf Reference”.

A tabela seguinte providencia uma breve descrição dos componentes que podem ser encontrados num perfil de user-agent:

Componente UAProf	Breve descrição
HardwarePlatform	<p>A componente <i>HardwarePlatform</i> engloba informação sobre as capacidades hardware do dispositivo móvel.</p> <p>Exemplos: capacidade cores, nome do modelo do dispositivo móvel, capacidade text input, tamanho do visor e capacidade áudio.</p>
SoftwarePlatform	<p>A componente <i>SoftwarePlatform</i> engloba informação sobre as características software do dispositivo móvel.</p> <p>Exemplos: codificadores áudio e video suportados, conjuntos de caracteres aceites, capacidade Java, tipos de conteúdo aceites / tipos MIME, sistema operativo e versão.</p>
NetworkCharacteristics	<p>A componente <i>NetworkCharacteristics</i> engloba informação relativa às capacidades de ligação à rede.</p> <p>Exemplos: protocolos suportados (CSD, GPRS, SMS, EDGE...) e métodos de encriptação suportados (WTLS, SSL, TLS...).</p>
BrowserUA	<p>A componente <i>BrowserUA</i> engloba informação relative ao browser do dispositivo móvel.</p> <p>Exemplos: nome e versão do browser, versão HTML suportada, versão XHTML suportada e capacidade JavaScript.</p>
WapCharacteristics	<p>A componente <i>WapCharacteristics</i> engloba informação sobre as características WAP suportadas pelo dispositivo móvel.</p> <p>Exemplos: capacidade DRM (digital rights management), valor máximo do deck WML, versão WAP suportada e bibliotecas</p>

PushCharacteristics	<p>WMLScript suportadas.</p> <p>A componente <i>PushCharacteristics</i> engloba informação sobre as capacidades WAP Push do dispositivo móvel.</p> <p>Exemplos: codificação de caracteres suportada, conjunto de caracteres suportados, tipos de conteúdo / tipo MIME suportados e valor máximo da mensagem WAP Push.</p>
MmsCharacteristics	<p>A componente <i>MmsCharacteristics</i> engloba informação sobre as capacidades MMS (Multimedia Messaging Service) do dispositivo móvel.</p> <p>Exemplos: valor máximo da mensagem MMS suportada, resolução máxima de imagens e conjunto de caracteres suportadas.</p>

Collection Types e Tipos de Atributos presentes no UAProf

No UAProf, todo o atributo está associado a um “collection type” e a um “attribute type”. Existem três “collection types” (*Simple*, *Bag*, *Seq*) e quatro “attribute types” (*Boolean*, *Dimension*, *Literal*, *Number*). A tabela seguinte apresenta uma breve descrição sobre estes tipos presentes no UAProf:

Collection type	Descrição
Simple	Um atributo do tipo collection <i>Simple</i> tem um único valor. Exemplo: atributo <i>ScreenSize</i> : <pre><prf:ScreenSize>208x208</prf:ScreenSize></pre>
Bag	Um atributo do tipo collection <i>Bag</i> possui uma lista de valores desordenados. Exemplo: atributo <i>CcppAccept-Charset</i> , define quais os conjuntos de caracteres suportados pelo dispositivo móvel: <pre><prf:CcppAccept-Charset> <rdf:Bag> <rdf:li>UTF-8</rdf:li> <rdf:li>ISO-10646-UCS-2</rdf:li> <rdf:li>ISO-8859-1</rdf:li></pre>

Seq	<pre> <rdf:li>US-ASCII</rdf:li> </rdf:Bag> </prf:CcppAccept-Charset> Um atributo do tipo collection Seq possui uma lista de valores ordenados. Exemplo: atributo <i>CcppAccept-Language</i>, que define o idioma preferencial do utilizador. Os itens são listados por ordem de preferência. <prf:CcppAccept-Language> <rdf:Seq> <rdf:li>en</rdf:li> <rdf:li>de</rdf:li> <rdf:li>fr</rdf:li> <rdf:li>zh-TW</rdf:li> <rdf:li>zh-CN</rdf:li> </rdf:Seq> </prf:CcppAccept-Language> </pre>
Attribute type	Descrição
Boolean	<p><i>Verdadeiro ou Falso.</i></p> <p>Exemplo: atributo <i>ColorCapable</i>:</p> <pre><prf:ColorCapable>Yes</prf:ColorCapable></pre>
Dimension	<p>Dois inteiros maiores ou iguais a zero com o character “x” entre eles.</p> <p>Exemplo: atributo <i>ScreenSize</i>:</p> <pre><prf:ScreenSize>208x208</prf:ScreenSize></pre>
Literal	<p>Uma string.</p> <p>Exemplo: atributo <i>Vendor</i>:</p> <pre><prf:Vendor>Nokia</prf:Vendor></pre>
Number	<p>Um inteiro maior ou igual a zero.</p> <p>Exemplo: atributo <i>Push-MsgSize</i>, que contém o valor máximo do tamanho da mensagem WAP Push:</p> <pre><prf:Push-MsgSize>1400</prf:Push-MsgSize></pre>

Capítulo 3

Arquitectura

Tal como explicitado no início do documento, o principal objectivo do trabalho é a correcta identificação de um dispositivo móvel e a consequente extracção das suas características. Para atingir este fim, a arquitectura deve ser desenhada tendo em conta alguns princípios:

- Abrangência dos dispositivos
- Focalização na identificação correcta dos dispositivos
- Optimização dos recursos disponíveis
- Entrega de conteúdo transversal à heterogeneidade dos dispositivos

Dada a enorme variedade de dispositivos móveis existente actualmente, convém que o sistema possua um repositório de dispositivos integrado. Será também vantajoso se esta base de dados de dispositivos for local em oposição a uma localização remota, já que a consequente redução de pedidos http só favorece a performance do sistema. Além de que a informação estará sempre disponível, o que não acontece no segundo caso numa eventual falha de conectividade da rede.

Adicionalmente, o sistema deve gerar conteúdo específico de acordo com as características do dispositivo. As duas condições atrás referidas implicam que o processamento desta informação não pode estar sujeito a condições de operação específicas como no caso da detecção do lado do cliente. Assim, foi determinado que a detecção do dispositivo e o processamento da informação devem residir no lado do servidor. Este facto trás vantagens em vários aspectos, nomeadamente:

- retira o processamento da informação do lado do cliente, o que em alguns casos implicaria execução de ficheiros de tamanho considerável e a inclusão de bibliotecas adicionais.
- possibilita o aumento da abrangência dos dispositivos através da separação da camada de apresentação do conteúdo dos dados.
- blinda a lógica programática o que favorece a correcta execução do código e melhora a performance geral do sistema.
- permite a utilização de cache para economizar recursos.

Para cumprir os requisitos propostos, foi escolhida a linguagem Java para a programação do módulo de software residente no servidor. Esta escolha é a óbvia visto que a grande percentagem de dispositivos móveis suporta a execução de código Java e a visualização de páginas JSP.

Tendo em conta os aspectos atrás mencionados, optou-se por utilizar o WURFL como base de dados de dispositivos residente no servidor Apache Tomcat v6.0.

Para o tratamento desse tipo de dados, a adopção do WURFL em oposição ao UAProf foi ponderada segundo algumas considerações das quais se citam algumas:

- O UAProf precisa de muita preparação e estudo no desenvolvimento da infra-estrutura que o suporta dentro de uma aplicação.
- Não existe garantia da precisão da informação dos perfis do UAProf (embora aconteça o mesmo com o WURFL, este oferece a possibilidade de corrigir eventuais discrepâncias através da edição do ficheiro patch).
- O WURFL possibilita a modelização de qualquer capacidade de um qualquer dispositivo independentemente dos fabricantes o que favorece a recolha de informação baseada na experiência de utilização.
- O WURFL pode ser instalado localmente num Web-site sem depender da recolha de dados duma base de dados remota.

-O WURFL tem uma comunidade de desenvolvimento open-source que actualiza com regularidade a lista de dispositivos suportados. Nesse processo é importada a informação dos perfis UAProf para as características mais básicas e à medida que forem descobertas diferenças, estas são corrigidas manualmente.

O módulo deve ser capaz de identificar correctamente o dispositivo e de lhe devolver as capacidades disponíveis no repositório. Para esse efeito o terminal deve enviar pedidos ao servidor para aceder a uma página onde lhe será apresentado as diversas capacidades. A descoberta dessas características fica a cargo de um *scriptlet* criado para usar a API Java do WURFL. Cada pedido enviado pelo cliente é processado pelo servidor e gera uma resposta de envio de uma página JSP personalizada com a linguagem de markup mais conveniente para o terminal móvel. Esta tarefa é levada a cabo pela biblioteca WALL instalada também no servidor.

Nesta arquitectura não está conceptualizado o conceito de agente para definir o módulo de software responsável pela interacção com o cliente. Essa abordagem permitiria certamente uma maior interacção com o terminal e uma adopção do conceito de contexto situacional com todas as vantagens daí inerentes. Contudo essa perspectiva trás problemas relativos à escalabilidade do sistema e necessita da implementação de uma solução sobre um Framework adaptado a esse efeito. Esse tipo de soluções não está completamente estabilizado e ainda necessita de standarts que recomendem a sua utilização por parte da indústria de comunicação móvel. As soluções pesquisadas nesse sentido de contexto situacional recaem fora do âmbito deste trabalho.

A simplicidade da arquitectura proposta visa acima de tudo propiciar robustez ao sistema e dotá-lo de uma performance considerável. Isto traduz-se na focalização na identificação acertada do dispositivo e na grande abrangência dos terminais. Além de que a distinção entre um browser móvel e um típico browser de um computador portátil ou de um PC fixo ser um requisito essencial.

Na figura seguinte é mostrada a arquitectura proposta:

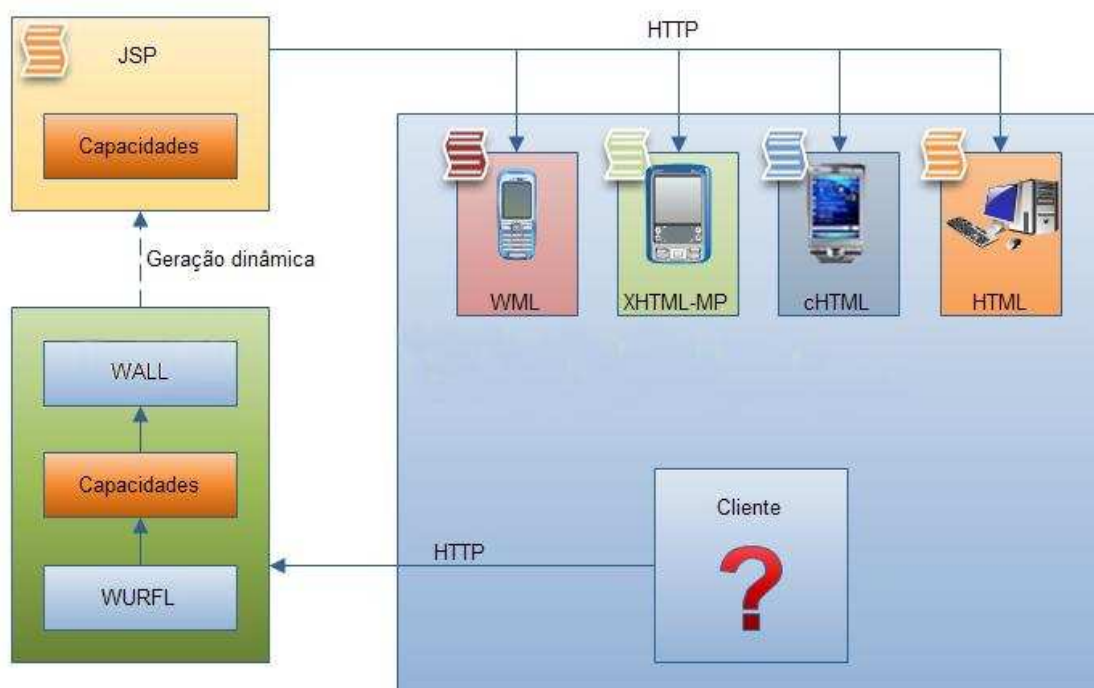


Figura 4. - Arquitectura proposta do sistema.

Capítulo 4

4.1 Implementação

De modo a desenvolver uma solução na arquitectura proposta foi necessário compreender a estrutura, funcionamento e sintaxe de desenvolvimento de aplicações Web no ambiente Java EE (Enterprise Edition).

Para esse efeito foi escolhida a ferramenta de desenvolvimento integrada Eclipse na versão Ganymede. Foi necessário efectuar várias configurações para dotá-la das bibliotecas e suporte requerido, nomeadamente:

- instalação da plataforma WTP (Web Tools Platform)
- instalação do servidor Java Apache Tomcat (versão 6.0)
- instalação das bibliotecas que oferecem o suporte JSTL (Java Server Pages Standart Tag Library) através do pacote Jakarta Taglibs.
- instalação do suporte à JVM, jdk versão 1.6.7.

Depois de efectuada a configuração com sucesso criou-se um novo projecto de desenvolvimento dinâmico Web. Este gerou automaticamente as pastas e ficheiros necessários ao seu correcto funcionamento juntamente com o ficheiro web.xml que é o ficheiro de configuração que define o modo de depleção de uma aplicação Web . As especificidades do contentor web da aplicação ficam correctamente definidas e em conformidade para execução com o servidor Tomcat.

Seguidamente, foi copiado o ficheiro wurfl.xml para o directório Web-Inf da aplicação juntamente com o ficheiro de actualização wurfl_patch.xml que contém os dispositivos identificados como sendo não móveis (browsers típicos de PC, i.e. Mozilla Firex Fox, Microsoft Internet Explorer, etc.)

Foi criada outra pasta dentro do mesmo directório com os ficheiros jars necessários à correcta interpretação e manipulação das páginas JSP por parte WURFL. Esses ficheiros são:

- jstl.jar
- standart.jar
- xom-1.0.jar
- wurfltags.jar

A biblioteca WALL fica disponível no sistema pela criação de outra pasta denominada “tld” e a respectiva inclusão na mesma do ficheiro wall.tld.

Após este passo, é necessário incluir as seguintes directrizes de pré-processamento no início de cada página JSP criada:

```
<%@ taglib uri="/WEB-INF/tld/wall.tld" prefix="wall" %>
```

```
<%@ taglib uri='http://java.sun.com/jstl/core' prefix='c' %>
```

A primeira directriz visa activar o reconhecimento e validação das tags específicas da biblioteca WALL dentro da sintaxe JSP.

A segunda directriz disponibiliza o suporte JSTL e activa o reconhecimento da sintaxe EL nas mesmas páginas JSP.

O ficheiro de descrição de depleção web.xml teve de ser alterado quer para suportar a sintaxe EL dentro da aplicação do projecto criado, quer para iniciar o servlet que recebe os pedidos de identificação de dispositivos dentro do WURFL:

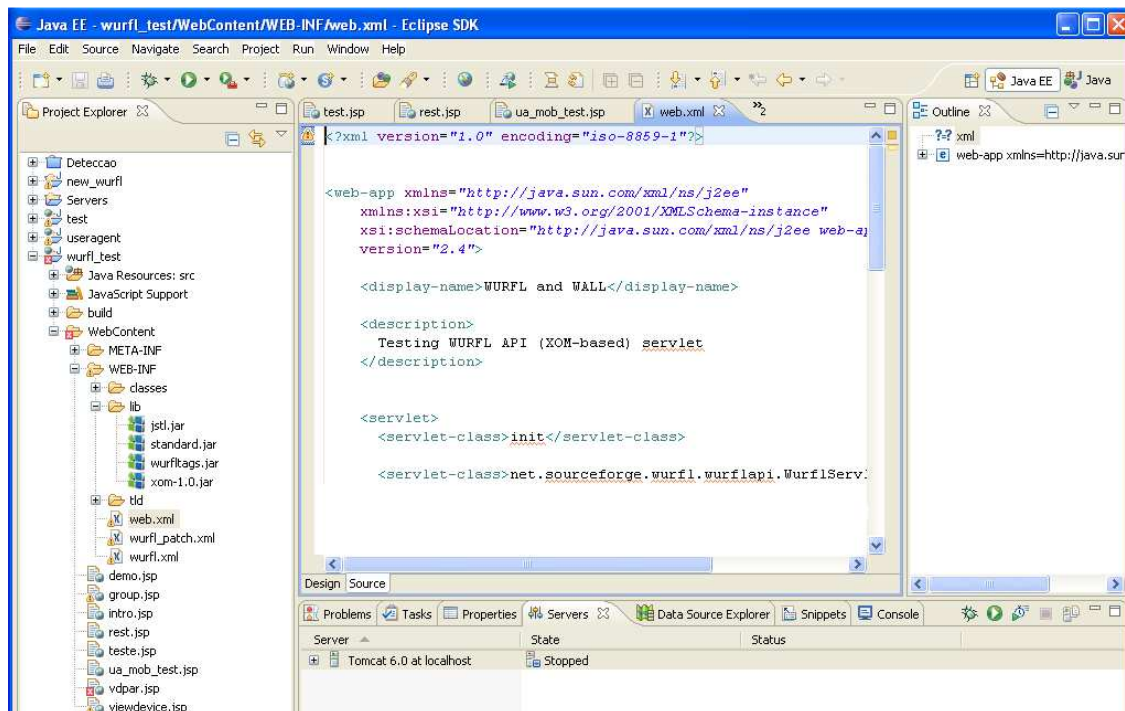


Figura 5. - Configuração do ficheiro web.xml.

Seguidamente, criaram-se páginas JSP para serem acedidas pelos browsers dos dispositivos com vista à identificação destes e à visualização das suas capacidades.

De início, foram criadas as várias páginas com sintaxe html mas respeitando as tags do WALL juntamente com código de scriptlets java para interagir com a API do WURFL.

O primeiro passo foi extrair o cabeçalho user-agent do pedido http e passá-lo como argumento à classe CapabilityTester.Java.

```
<%
    httpRequest = request.getHeader("user-agent");
    if (httpRequest != null){
        cap = new wurfl_p.CapabilityTester(httpRequest);
        id = cap.device_id;
    }
%>
```

Esta classe tem como propósito efectuar um teste preliminar na identificação do dispositivo através da instanciação das classes UAManager e CapabilityMatrix por motivos de performance e aproveitamento da cache.

Uma vez retornado o device id do dispositivo é possível utilizar os métodos disponibilizados pela API do WURFL para inquirir o WURFL das capacidades do dispositivo identificado:

Segue o código que implementa a classe CapabilityTester:

```
package wurfl_p;

import net.sourceforge.wurfl.wurflapi.CapabilityMatrix;
import net.sourceforge.wurfl.wurflapi.ObjectsManager;
import net.sourceforge.wurfl.wurflapi.UAManager;

public class CapabilityTester {

    public String device_id, real, markup, access_key_support, table_support,
    css_cell_support, menu_with_select_element_support;
    public boolean isReal;

    public CapabilityTester (String httpRequest)
    {

        /** Passo 3 **/
        UAManager uaManager = ObjectsManager.getUAManagerInstance();

        /** Passo 4 **/
        device_id = uaManager.getDeviceIDFromUALoose(httpRequest);

        /** Passo 5 **/
        CapabilityMatrix capabilityMatrix =
        ObjectsManager.getCapabilityMatrixInstance();

        /** Passo 6 **/

        isReal =
        capabilityMatrix.isCapabilityDefinedInDevice(device_id,"actual_device_root");
        real = new Boolean(isReal).toString() ;
    }
}
```

```

        markup =
capabilityMatrix.getCapabilityForDevice(device_id,"preferred_markup");
        access_key_support =
capabilityMatrix.getCapabilityForDevice(device_id,"access_key_support");
        table_support =
capabilityMatrix.getCapabilityForDevice(device_id,"xhtml_supports_table_for_layout");
        css_cell_support =
capabilityMatrix.getCapabilityForDevice(device_id,"xhtml_supports_css_cell_table_coloring");
        menu_with_select_element_support
=capabilityMatrix.getCapabilityForDevice(device_id,"menu_with_select_element_recommen
ed");

        System.out.println ("\nUser Agent : "+httpRequest);
        System.out.println ("\nDevice ID : "+device_id);
        System.out.println ("\nReal Device : "+real);
        System.out.println ("preferred_markup : "+markup);
        System.out.println ("access_key_support : "+access_key_support);
        System.out.println ("menu_with_select_element_recommended:
"+menu_with_select_element_support);
        System.out.println ("xhtml_supports_table_for_layout : "+table_support);
        System.out.println ("xhtml_supports_css_cell_table_coloring:
"+css_cell_support);
    }
}

```

Para visualizar a lista de todas as capacidades do dispositivo, é passado como argumento o device id do dispositivo para outra página JSP através de um link no visor do dispositivo. Para esse efeito é usada a *anchor tag* da biblioteca WALL.:

```
<wall:a href="viewdevice.jsp?id=${id}">Ver detalhes</wall:a>
```

As capacidades estão agrupadas nos grupos predefinidos do WURFL pela tag <group>. Assim, para determinar o valor de uma capacidade do dispositivo basta seleccionar o link do grupo de capacidades correspondente. Todos os valores das capacidades definidas dentro desse grupo são devolvidos ao utilizador na página JSP visitada.

Nessa página é instanciado um vector que contém os valores das capacidades para esse grupo específico:

```
<%  
    Iterator capa_keys = cap_list.iterator();  
    while ( capa_keys.hasNext() ) {  
        String capa_key = (String) capa_keys.next();  
        String value = cm.getCapabilityForDevice(id, capa_key);  
        color = (i++ % 2 == 0) ? "#DDDDDD" : "white";  
    }  
>%
```

A escolha das capacidades relevantes para o utilizador deve ser tomada tendo em conta o grupo em que se inserem. Na secção Anexos deste documento estão documentadas todas as capacidades disponibilizadas para um dispositivo específico através desta implementação do WURFL.

Para efeitos de simplificação de utilização foram criadas duas páginas jsp chamadas hardware.jsp e software.jsp com as características mais relevantes do dispositivo. Caso uma capacidade seja suportada é devolvido o valor lógico *true* e *false* em caso contrário. Certas capacidades têm um valor inteiro associadas e no caso de não o dispositivo não ter um valor definido para uma capacidade é retornada a string *null*.

4.2 Resultados obtidos

O teste da aplicação desenvolvida foi dividido em dois passos:

- 1º- Teste em ambiente localhost com simuladores/emuladores de dispositivos móveis
- 2º- Teste em ambiente remoto com os mesmos simuladores/emuladores e depois com dispositivos reais.

Os simuladores/emuladores usados que efectuam os pedidos http ao servidor da aplicação são:

- Openwave Phone Simulator v7.0
- Nokia 3300 SDK v1.0

O simulador da Nokia requer porém a instalação de uma gateway WAP para a interpretação do tráfego http gerado. Para cumprir esse requisito foi instalado o software Nokia Gateway Simulator v4.0.

1º Testes em ambiente localhost:

Openwave Phone Simulator



Figura 6. - Identificação do dispositivo



Figura 7. - Hardware



Figura 8. - Hardware (cont.)



Figura 9. - Hardware (cont.)



Figura 10. - Hardware (cont.)



Figura 11. - Software



Figura 12. - Software (cont.)



Figura 13. - Software (cont.)



Figura 14. - Software (cont.)

Nokia 3300 SDK:



Figura 15. - Identificação do dispositivo

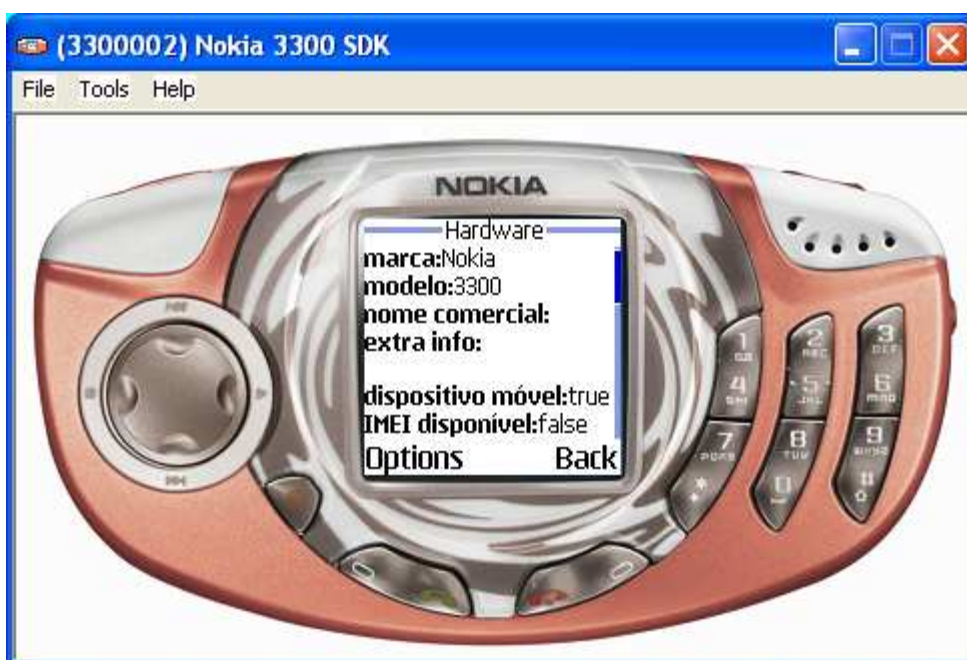


Figura 16. - Hardware



Figura 17. - Hardware (cont.)



Figura 18. - Hardware (cont.)



Figura 19. - Software

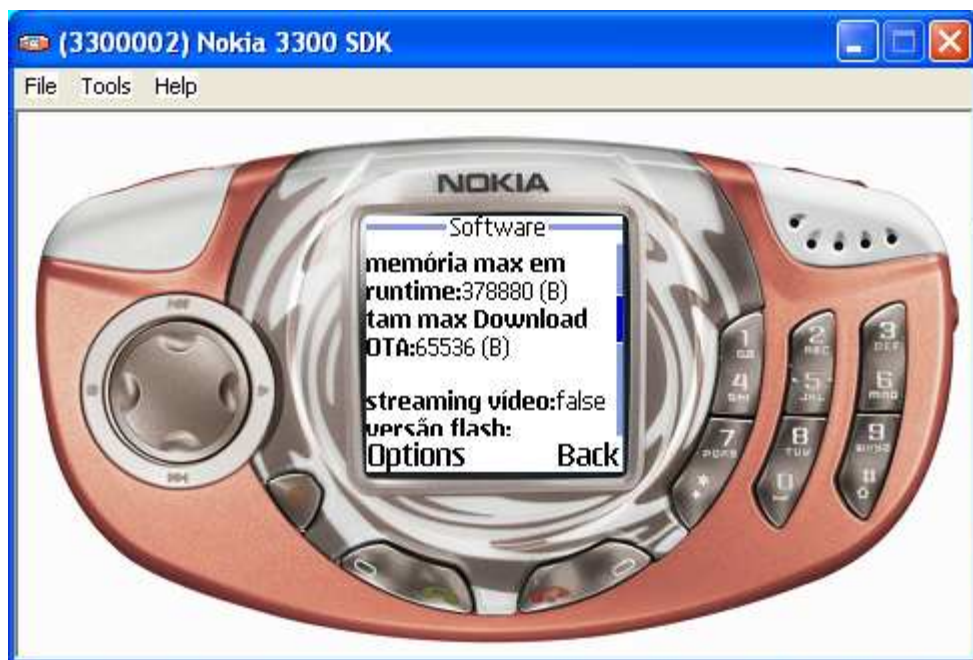


Figura 20. - Software (cont.)

Apesar de o simulador da Openwave não ter valores definidos para as capacidades de memória máxima em runtime e de tamanho máximo do ficheiro descarregado OTA (Over The Air), estas estão definidas para o dispositivo da Nokia devido à sua comunicação com a WAP Nokia gateway instalado no localhost.

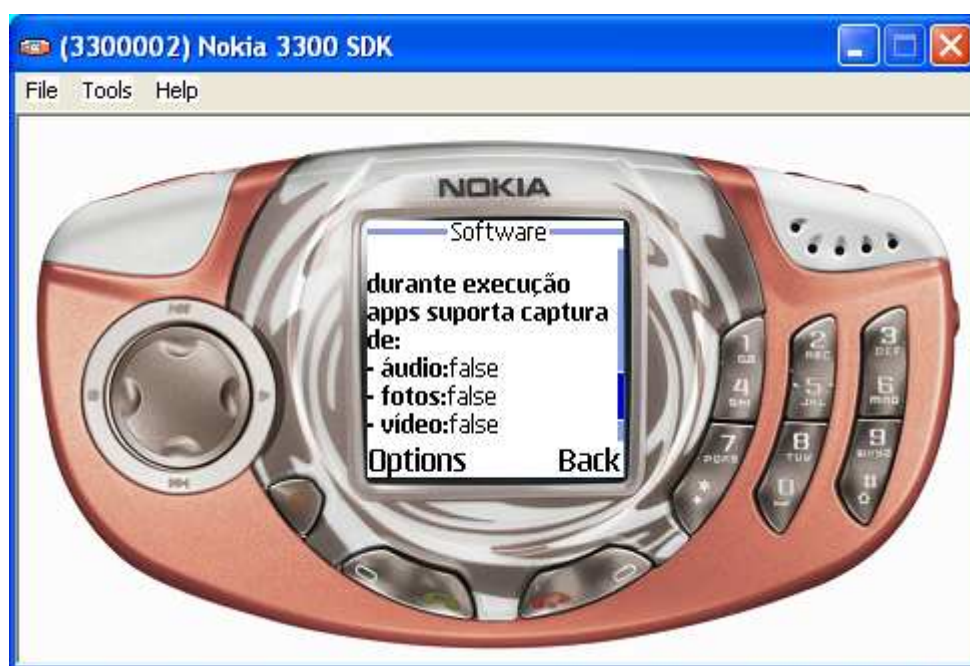


Figura 21. - Software (cont.)

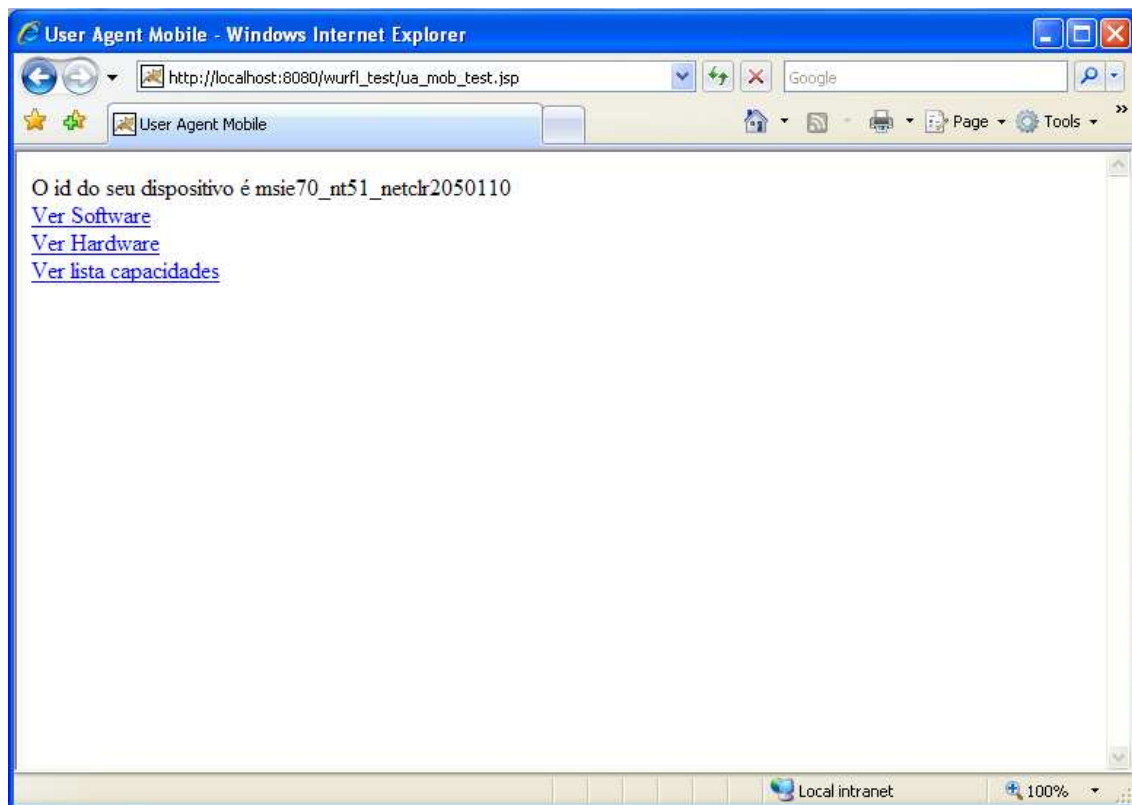


Figura 22. - Identificação de dispositivo não móvel (Internet Explorer de laptop).

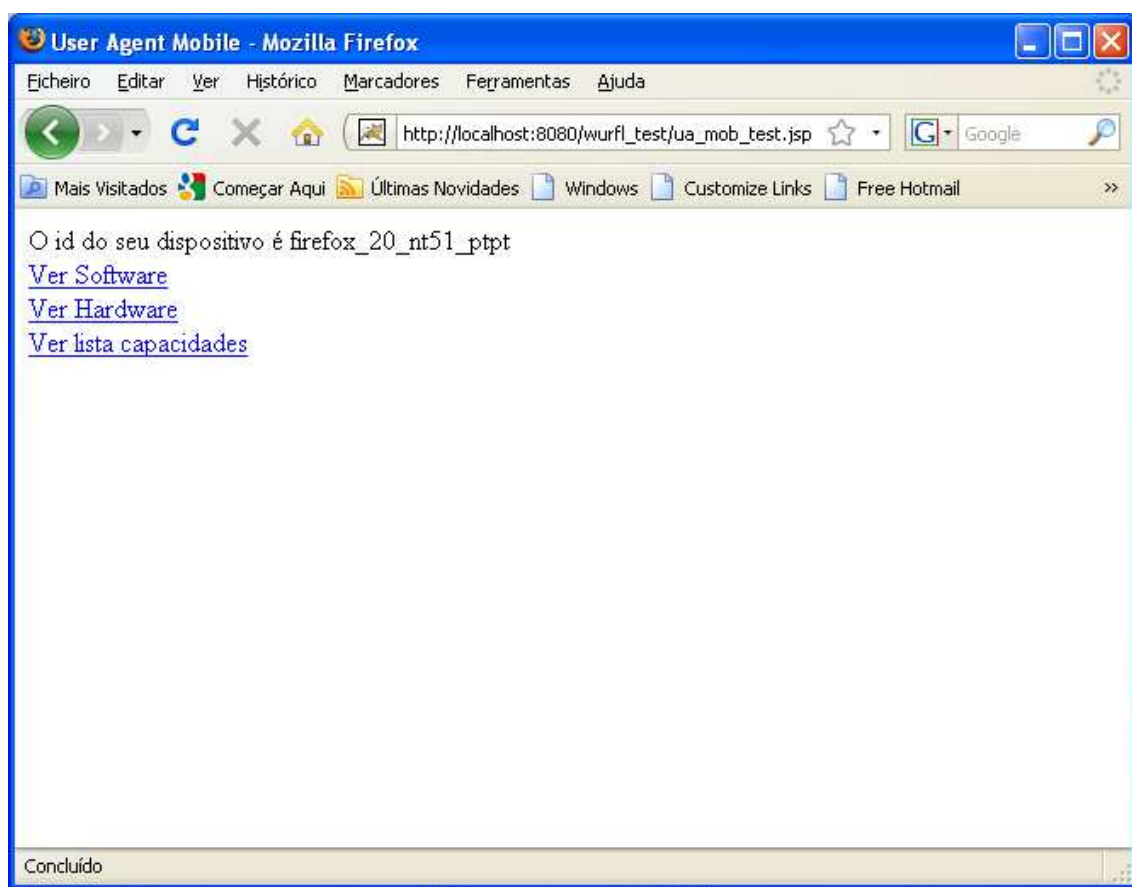


Figura 23. - Identificação de dispositivo não móvel (Mozilla Firefox de laptop).

Os browsers Internet Explorer e Mozilla Firefox foram também correctamente identificados e classificados como dispositivos não móveis.

2º Testes em ambiente remoto:

A instalação da aplicação no servidor tomcat v5.0 do yoda.fe.up.pt não permitiu executar os testes como esperado devido à compliance Java 1.4 exigida por este para compilar as classes da aplicação.

Capítulo 5

5.1 Conclusões

A implementação de um repositório local de dispositivos permitiu identificar e extrair as capacidades de um dispositivo móvel através da pesquisa do seu user-agent numa base de dados. A aplicação conseguiu distinguir correctamente os dispositivos que acedem à dita aplicação particularmente se são móveis ou não. As principais características são apresentadas e também é possível explorar todas as capacidades pela lista de capacidades.

As páginas jsp visualizadas foram geradas com markup dinâmico correspondente ao markup preferido de ambos os dispositivos móveis, xhtml-mp. Esta característica é uma mais-valia pois beneficia o desenvolvimento de aplicações válidas para a grande generalidade dos terminais móveis e também de modelos anteriores. Apesar de o sistema ser eficiente na identificação dos dispositivos não é infalível, visto que a alteração do campo do user-agent header contido no pedido http de um browser pode originar falsos positivos vindos de browsers de desktops. No entanto, visto que a maioria dos dispositivos móveis possuem esse campo protegido pela implementação do software do operador móvel, revela-se vantajoso numa aplicação que perspetive um número considerável de ligações de clientes. É também possibilitada a inclusão de um módulo que efectue as operações de extracção de capacidades dinâmicas dos dispositivos. Esse módulo, criado via JME e a executar no cliente deverá incluir as API's Java MMAPi (JSR-135) e Event Tracking API (JSR 190). A primeira permite a aquisição de sinais áudio e vídeo pelo dispositivo e posterior transmissão na rede. A segunda permite a monitorização de eventos definidos pelo gestor do sistema. Estes deverão contemplar a capacidade de memória de execução e de armazenamento livres bem como a capacidade restante da bateria. A criação de perfis associados aos dispositivos dos utilizadores permite a associação de contexto situacional ao sistema criado.

5.2 Perspectivas de desenvolvimento

A aquisição de informação sobre aplicações instaladas no dispositivo e também de sinais de periféricos requer a criação de um módulo programado em JME (Java Micro Edition) e sua instalação no cliente. A comunicação deste último com a aplicação desenvolvida poderá auxiliar eventuais tomadas de decisão relacionadas com o contexto operacional. As capacidades devolvidas pelo WURFL poderão facilitar a manipulação de uma API específica do JME para obtenção de sinais periféricos do terminal.

A manipulação das API's Java MMAPAPI (JSR-135) e Event Tracking API (JSR 190) no sistema juntamente com a criação de perfis de utilização dotarão o sistema da capacidade de responder a eventos relacionados com contexto-situacional.

Anexos

Lista de capacidades disponíveis no WURFL:

Grupo: **product_info** (informação genérica incluindo o nome da marca e modelo)

<i>Nome Capacidade</i>	Tipo	Descrição
brand_name	String	Marca (ex: Nokia)
model_name	String	Modelo (ex: N95)
marketing_name	String	Além da Marca e Modelo alguns dispositivos possuem um nome de marketing (ex: BlackBerry 8100 Pearl , Nokia 8800 Scirocco , Samsung M800 Instinct).
model_extra_info	String	Além da Marca e Modelo (e possivelmente um nome de marketing), alguns dispositivos podem conter informação extra (ex: Nokia N95 8GB , Sharp 902SH Vodafone).
unique	verdadeiro/falso	UA é repetido para dispositivos diferentes

		(ocorrência rara mas fatal)
un uniqueness_handler	String	Como tratar uma string de UA não única (API/framework pode disponibilizar algum mecanismo para tratar pedidos http directamente).
is_wireless_device	verdadeiro/falso	Especifica se o dispositivo é wireless ou não. Tipicamente, um telemóvel ou um PDA são considerados wireless. Um PC desktop ou laptop não são.
device_claims_web_support	verdadeiro/falso	Dependendo se é wireless ou não, o browser pode especificar suporte web ou não. O browser Opera para Symbian é um exemplo de um browser que especifica suporte web. Neste caso, o browser tenta efectuar a renderização de uma página desenvolvida para apresentação web independentemente do dispositivo.
has_pointing_device	verdadeiro/falso	(obsoleto) : Se o dispositivo possui uma pen/stylus para servir de interacção com o visor.
pointing_method	joystick, stylus, touchscreen, clickwheel, "" (empty string)	Reflecte a capacidade activar links e widgets por via de uma pen, pelo dedo, por um joystick ou por uma click Wheel estilo BlackBerry. Se esta capacidade retorna a empty string (""), a capacidade anterior (has_pointing_device) deverá ser false.
has_qwerty_keyboard	verdadeiro/falso	Alguns dispositivos apresentam um teclado 'qwerty'. Esta capacidade pode influenciar a implementação de formulários e outras funções. Teclados virtuais do tipo Palm são suficientes para tornar a capacidade verdadeira.
can_skip_aligned_link_row	verdadeiro/falso	Muitos dispositivos recentes (ou combinações browser/dispositivo)

		<p>permitem ao utilizador saltar uma linha inteira de links (por ex: <u>link1</u> <u>link2</u> <u>link3</u> <u>link4</u>) com apenas um click para baixo. Outros dispositivos forçam o utilizador a clicar múltiplas vezes para saltar a lista de links.</p>
uaprof,uaprof2,uaprof3	String (URL)	Os urls UAProf são tipicamente extraídos do cabeçalho HTTP. Esta capacidade pode ser útil quando tal operação não está disponível por qualquer razão.
nokia_series	Inteiro	Nokia Series 20/30/40/60/80 ou 90
nokia_edition	Inteiro	Plataforma de Desenvolvimento (1/2/3/...)
device_os	String	Informação sobre o SO do dispositivo
device_os_version	String	Versão do SO do dispositivo
mobile_browser	String	Informação sobre o browser do dispositivo (Openwave, Nokia, Opera, Access, Teleca,...)
device_os_version	String	Versão do browser

Grupo: **wml_ui** (User Interface para WML browser)

<i>Nome da capacidade</i>	Tipo	Descrição
proportional_font	verdadeiro/falso	A fonte standart é proporcional.
built_in_back_button_support	verdadeiro/falso	Permite ao utilizador clicar num botão para voltar atrás.
card_title_support	verdadeiro/falso	O dispositivo apresenta o título da página no ecrã.
softkey_support	verdadeiro/falso	Softkeys são suportadas.
table_support	verdadeiro/falso	O browser renderiza tabelas formatadas "correctamente". (Em vez de uma célula por linha)
numbered_menus	verdadeiro/falso	O browser lista números por forma a escolher um elemento de uma lista directamente.
menu_with_select_element_recommended	verdadeiro/falso	Um elemento "select" é o formato adequado para menus.
menu_with_list_of_links_recommended	verdadeiro/falso	Uma lista de "links" é o formato adequado para menus.
icons_on_menu_items_support	verdadeiro/falso	Links podem estar associados a icons.
break_list_of_links_with_br_element_recommended	verdadeiro/falso	Na apresentação de uma lista de "links" é sugerido o uso da tag (quebra de linha).
access_key_support	verdadeiro/falso	Respeita o atributo "accesskey" da tag anchor.
wrap_mode_support	verdadeiro/falso	O browser pode ser forçado a juntar linhas ou não.
times_square_mode_support	verdadeiro/falso	"time_square_mode_support" é uma capacidade específica do browser Openwave relacionada com código que apresente o modo "nowrap".

deck_prefetch_support	verdadeiro/falso	Suporte à pré-captura de decks.
elective_forms_recommended	verdadeiro/falso	Elementos "input" e "select" podem/devem ser colocados num único "card" em vez de vários.
wizards_recommended	verdadeiro/falso	Recomenda a utilização de wizards.
image_as_link_support	verdadeiro/falso	Possibilidade de usar imagens como "link".
insert_br_element_after_widget_recommended	verdadeiro/falso	É recomendado o uso de quebra de linha após um widget.
wml_can_display_images_and_text_on_same_line	verdadeiro/falso	Possibilidade de mostrar imagens e texto na mesma linha.
wml_displays_image_in_center	verdadeiro/falso	Alguns dispositivos apresentam imagens alinhadas ao centro por defeito.
opwv_wml_extensions_support	verdadeiro/falso	Capacidade específica dos browsers Openwave e possivelmente de terceiros compatíveis com estes. Extensões WML referem-se a pictogramas e outras tags que não foram normalizadas pelo WAP forum.
wml_make_phone_call_string	string	Prefixo para iniciar uma chamada de voz.

Grupo: **chtml_ui** (User Interface para browsers que suportam Compact HTML em i;Mode)

<i>Nome da capacidade</i>	Tipo	Descrição
chtml_display_accesskey	verdadeiro/falso	O dispositivo apresenta um número quando se utiliza uma "accesskey".
emoji	verdadeiro/falso	Emoji são caracteres especiais que aparecem em páginas i-Mode como ícones pequenos.
chtml_can_display_images_and_text_on_same_line	verdadeiro/falso	Tal como no WML, esta capacidade é verdadeira se o dispositivo consegue apresentar imagens e texto na mesma linha.
chtml_displays_image_in_center	verdadeiro/falso	Tal como no WML, esta capacidade é verdadeira se o dispositivo alinha imagens ao centro por defeito.
imode_region	string	Este campo descreve a região de comercialização do dispositivo, útil para emoji's e outros. Alguns valores possíveis: "ja" para Japão, "eu" para Europa, "us" para USA "as" for Ásia. "none" significa desconhecido ou não especificado.
chtml_make_phone_call_string	string	Prefixo para iniciar uma chamada de voz.
chtml_table_support	verdadeiro/falso	As tabelas não são suportadas por defeito em dispositivos i-mode/chtml. Esta capacidade especifica o suporte destas tabelas para este tipo de dispositivos.

Grupo: **xhtml_ui** (User Interface para browsers HTML/XHTML-MP)

<i>Nome da capacidade</i>	Tipo	Descrição
xhtml_honors_bgcolor	verdadeiro/falso	Podem ser definidas cores de background.
xhtml_supports_forms_in_table	verdadeiro/falso	Podem ser especificados formulários dentro de uma tabela.
xhtml_support_wml2_namespace	verdadeiro/falso	A especificação WML version 2.0 é suportada. O dispositivo consegue renderizar conteúdo WML 2.0 correctamente.
xhtml_autoexpand_select	verdadeiro/falso	Alguns dispositivos expendem elementos "select" automaticamente. (MOT T720, por ex.)
xhtml_select_as_dropdown	verdadeiro/falso	O dispositivo apresenta elementos "select" como listas dropdown.
xhtml_select_as_radiobutton	verdadeiro/falso	O dispositivo apresenta elementos "select" como botões "radio".
xhtml_select_as_popup	verdadeiro/falso	O dispositivo apresenta elementos "select" como listas popup.
xhtml_display_accesskey	verdadeiro/falso	O dispositivo apresenta um número quando é utilizada uma "accesskey".
xhtml_supports_invisible_text	verdadeiro/falso	
xhtml_supports_inline_input	verdadeiro/falso	Alguns browsers permitem a escrita de texto no cursor. Outros obrigam à escrita

		noutra estrutura de entrada de dados.
xhtml_supports_monospace_font	verdadeiro/falso	Verdadeiro se somente suporta fontes "monospace".
xhtml_supports_table_for_layout	verdadeiro/falso	O suporte anunciado para tabelas é suficientemente robusto para utilizar tabelas para apresentar conteúdo em layout no ecrã. (Utilizada pela biblioteca WALL)
xhtml_supports_css_cell_table_coloring	verdadeiro/falso	O suporte CSS é suficientemente robusto para utilizar cores (quando definidas por CSS) correctamente em tabelas. (Utilizada pela biblioteca WALL)
xhtml_format_as_css_property	verdadeiro/falso	Verdadeiro se o dispositivo suporta uma propriedade CSS para definir o formato de um campo "input". (Utilizada pela biblioteca WALL)
xhtml_format_as_attribute	verdadeiro/falso	Verdadeiro se o dispositivo suporta o atributo "format" em campos "input". (Utilizada pela biblioteca WALL)
xhtml_nowrap_mode	verdadeiro/falso	Verdadeiro se o dispositivo suporta o modo "wrap" como atributo XHTML. (Utilizada pela biblioteca WALL)
xhtml_marquee_as_css_property	verdadeiro/falso	Verdadeiro se o dispositivo suporta "MARQUEE" através da sintaxe CSS. (Utilizada pela biblioteca WALL)

xhtml_readable_background_color1	string	Esta capacidade e a seguinte permitem definir duas cores que operam visualmente no dispositivo. Por exemplo, ao utilizar uma destas duas cores como background é prevenido o desaparecimento de hiperligações em relação ao background. (Utilizada pela biblioteca WALL)
xhtml_readable_background_color2	string	Ver capacidade anterior. (Utilizada pela biblioteca WALL)
xhtml_allows_disabled_form_elements	verdadeiro/falso	XHTML permite definir elementos "form" não activados.
xhtml_document_title_support	verdadeiro/falso	Embora seja possível definir o título do documento pela tag <title>, nem todos os dispositivos efectuem a sua renderização. Esta capacidade descreve esse comportamento. (Utilizada pela biblioteca WALL)
xhtml_preferred_charset	string	UTF-8 deve ser suportado por todos os dispositivos por defeito mas alguns apresentam problemas. Nesta capacidade são propostos "charsets" alternativos tais como iso8859. Este campo não apresenta o "charset" exacto devido à necessidade de especificação da região ISO do utilizador. (1, 15, outros)
opwv_xhtml_extensions_support	verdadeiro/falso	Esta capacidade referencia o

		suporte a extensões disponibilizadas pelo browser da Openwave. (Utilizada pela biblioteca WALL)
xhtml_make_phone_call_string	string	Prefixo utilizado para iniciar uma chamada de voz. (Utilizada pela biblioteca WALL)
xhtmlmp_preferred_mime_type	string	A maioria dos dispositivos devem suportar text/html e um tipo MIME específico do XHTML MP. Esta capacidade referencia o tipo MIME que deverá funcionar para o dispositivo. A predefinição é <i>text/html</i> .
xhtml_table_support	verdadeiro/falso	As tabelas deverão sempre ser suportadas como sintaxe mas muitas vezes a renderização deixa a desejar. Esta capacidade indica se o dispositivo é capaz de efectuar renderização de tabelas razoavelmente.
xhtml_supports_file_upload	verdadeiro/falso	Obsoleta. Ver a capacidade xhtml_file_upload.
xhtml_send_sms_string	none, sms:,smsto:	Indica se o dispositivo suporta a sintaxe href="sms:+num" para activar o serviço SMS do cliente através de um link. A sintaxe também poderá ser "smsto:" em alguns dispositivos ou não suportada.
xhtml_send_mms_string	none, mms:,mmsto:	Indica se o dispositivo suporta a sintaxe href="mms:+num"

		para activar o serviço MMS do cliente através de um link. A sintaxe também poderá ser "mmsto:" em alguns dispositivos ou não suportada.
xhtml_file_upload	not_supported, supported, supported _user_intervention	Indica se o browser respeita o elemento <code>type="file"</code> em formulários (utilizadores podem fazer upload de ficheiros do seu dispositivo para um servidor remoto). Em alguns dispositivos, os utilizadores terão que copiar/mover o ficheiro de um dado directório para um directório visível ao Web browser no sistema de ficheiros do dispositivo.
xhtml_supports_iframe	none,partial,full	Indica se o browser suporta iFrame. Suporte parcial indica que o conteúdo na página referida não é renderizado se colocado após o iframe.
cookie_support	verdadeiro/falso	Indica se o browser suporta cookies (tenha atenção que a cookie poderá estar em falta no caso de o operador a remover. Do mesmo modo, um dispositivo sem suporte para cookies pode activar automaticamente este service no caso de uma WAP gateway tartar as cookies do lado do cliente.
accept_third_party_cookie	verdadeiro/falso	Indica se o dispositivo aceita uma cookie enviada por um

		pixel numa página de um domínio diferente (assumindo as configurações predefinidas no dispositivo).
--	--	---

Grupo: **css** aspectos relacionados com CSS

<i>Nome da capacidade</i>	Tipo	Descrição
css_supports_width_as_percentage	verdadeiro/falso	Identifica os dispositivos cuja propriedade CSS não funciona como esperado (caso de muitos Nokias). Nesse caso, é recomendado usar a capacidade max_image_width para forçar o tamanho correcto das tabelas ao longo do ecrã.

Grupo: **ajax** suporte para mobile ajax

<i>Nome da capacidade</i>	Tipo	Descrição
ajax_support_javascript	verdadeiro/falso	Um dispositivo diz-se Javascript enabled somente se os seguintes recursos são confiavelmente suportados: alert, confirm, access form elements (dynamically set/modify values), setTimeout, setInterval, document.location. Se o dispositivo falhar um destes testes, a capacidade deverá ser marcada como falso.
ajax_support_getelementbyid	verdadeiro/falso	Consegue seleccionar um nó através do seu ID.
ajax_xhr_type	none, standard, msxml2, legacy_microsoft	Qual a sintaxe seguida para criar um objecto XMLHttpRequest(): none ou XMLHttpRequest(); ActiveXObject("Msxml2.XMLHTTP") e ActiveXObject("Microsoft.XMLHTTP")
ajax_support_inner_html	verdadeiro/falso	Consegue colocar um segmento de códigoHTML dentro de uma DIV. No caso de documentos XML, poderá não funcionar como esperado (principalmente no iPhone). Para evitar esta situação convém usar o tipo MIME/DTD correcto para os documentos se a intenção é usar innerHTML.
ajax_manipulate_dom	verdadeiro/falso	Indica se parentNode, getElementsByTagName e getElementByName são suportados.(todos eles).
ajax_manipulate_css	verdadeiro/falso	Capacidade de modificar o CSS programaticamente e conseguir que as

		mudanças sejam imediatamente reflectidas visualmente na renderização.
ajax_support_events	verdadeiro/falso	Indica se existe suporte para onload, onclick, onsubmit e onselect.
ajax_support_event_listener	verdadeiro/falso	(Event Listeners) Indica se o browser permite o registo de event listeners em event targets.

Grupo: **markup** linguagens de marcação suportadas

Nome da capacidade	Tipo	Descrição
xhtml_support_level	[-1 .. 4]	<p>Assumindo que o dispositivo suporta alguma forma de XHTML, esta capacidade mede o grau de confiança que certos recursos de design podem ser suportados, segundo a seguinte descrição:</p> <ul style="list-style-type: none"> - nível "-1": Não existe suporte XHTML de tipo algum. Possivelmente suporta WML. Ex. dispositivos Nokia 7110, 7210 e também todos os dispositivos UP.Browser 4 e 5. - nível "0" : Suporte XHTML básico. Largura do visor mínima: 100 pixel Suporte CSS inexistente ou não confiável. Fraco suporte para tabelas ou inexistente. Formulários básicos: text field, select-option, submit button. Poderá não suportar mascaras de input fields. Ex. dispositivo: SonyEricsson T610/T616 - nível "1" : Suporta XHTML com algum CSS. Largura do visor mínima: 120 pixels. As hiperligações poderão não ser coloridas via CSS. Suporte para tabelas básico: 2x2 ou mais. Colspan e rowspan poderão não ser suportados. "width" expressa em percentagem poderá ser não confiável. Ex. dispositivo: Sharp GX 10, Nokia 3650 - nível "2" : Assumem-se as mesmas capacidades do nível "1", mas poderão variar no futuro. Ex. dispositivo: Nokia6600,Nokia5300 - nível "3" : Excelente suporte CSS. Padding, border e margin são aplicados correctamente. Consegue aplicar cores a links, texto and background. Gráficos CSS são perfeitos ao nível do pixel.

		<p>Largura do visor mínima: 164 pixels.</p> <p>Suporta tabelas complexas (mas não necessariamente tabelas nested) até 4 células numa fila.</p> <p>Ao definir o "font-size" em 10px ou mais produz texto legível.</p> <p>Suporta imagens de background inclusive quando aplicadas via CSS.</p> <p>Generalmente disponível em dispositivos 3G.</p> <p>Ex: Nokia Series 60 DP 3, Browsers: Openwave Mobile Browser 6.2, MS Mobile Explorer, BlackBerrys recentes. Netfront 3.2, Sony PSP...</p> <p>(assumindo largura de banda elevada, Edge/UMTS/...)</p> <p>- nível "4" :</p> <p>Nível 3 + suporte Ajax.</p> <p>Permite alterar a propriedade de display.</p> <p>XMLHttpRequest().</p> <p>Ex: Browser Safari (incluindo iPhone), MS Mobile Explorer, Openwave 7.2.</p>
preferred_markup	string	<p>Esta campo identifica a linguagem de markup mais adequada para o dispositivo. Esta capacidade é validada através das experiências dos vários contribuidores do WURFL e é utilizada pela biblioteca WALL.</p> <p>Valores para esta capacidade são semelhantes a: <i>wml_1_1</i>, <i>html_wi_imode_compact_generic</i> and <i>html_wi_oma_xhtmlmp_1_0</i>. Tal como qualquer outra capacidade este valor pode ser sobreposto por outro especificado no ficheiro patch.</p>
wml_1_1	verdadeiro/falso	Suporta WML versão 1.1
wml_1_2	verdadeiro/falso	Suporta WML versão 1.2
wml_1_3	verdadeiro/falso	Suporta WML versão 1.3
html_wi_w3_xhtmlbasic	verdadeiro/falso	XHTML basic é XHTML reduzido a um conjunto mínimo de tags e foi introduzido para servir de markup

		base que funcionasse em dispositivos com capacidades muito limitadas.
html_wi_oma_xhtmlmp_1_0	verdadeiro/falso	XHTML MP é XHTML basic com a adição de tags extra para permitir a aplicação de WCSS (tag e atributo 'style' e a tag 'hr')
html_wi_imode_html_1	verdadeiro/falso	Suporta DoCoMo's iHTML versão 1.0
html_wi_imode_html_2	verdadeiro/falso	Suporta DoCoMo's iHTML versão 2.0
html_wi_imode_html_3	verdadeiro/falso	Suporta DoCoMo's iHTML versão 3.0
html_wi_imode_html_4	verdadeiro/falso	Suporta DoCoMo's iHTML versão 4.0
html_wi_imode_html_5	verdadeiro/falso	Suporta DoCoMo's iHTML versão 5.0
html_wi_imode_htmlx_1	verdadeiro/falso	Suporta DoCoMo's xHTML versão 1.0
html_wi_imode_compact_generic	verdadeiro/falso	Suporta compact HTML (cHTML) genérico.
html_web_3_2	verdadeiro/falso	Suporta HTML versão 3.2
html_web_4_0	verdadeiro/falso	Suporta HTML versão 4
voicexml	verdadeiro/falso	Suporta voice XML.
multipart_support	verdadeiro/falso	Suporta conteúdo multipart.

Grupo: **cache**

<i>Nome da capacidade</i>	Tipo	Descrição
total_cache_disable_support	verdadeiro/falso	Possibilita a desactivação completa da cache do browser.
time_to_live_support	verdadeiro/falso	Se o dispositivo suporta 'time to live'(TLL) ou não. TTL é o período de tempo que um dispositivo armazena um deck em cache. A predefinição para browsers Openwave é 30 dias (ou até a memória esgotar). No caso de um deck conter informação sensível ao tempo é possível especificar um TTL mais curto de modo a que o dispositivo recarregue o deck do servidor com maior frequência.

Grupo: **display**

<i>Nome da capacidade</i>	Tipo	Descrição
resolution_width	Qualquer número inteiro	Este campo representa a largura utilizável do visor do dispositivo expressa em pixels.
resolution_height	Qualquer número inteiro	Este campo representa a altura utilizável do visor do dispositivo expressa em pixels.
columns	Qualquer número inteiro	Número de colunas apresentado.
rows	Qualquer número inteiro	Número de linhas apresentado.
max_image_width	Qualquer número inteiro	Largura das imagens visíveis expressa em pixels.
max_image_height	Qualquer número inteiro	Altura das imagens visíveis expressa em pixels.

Grupo: **image_format**

<i>Nome da capacidade</i>	Tipo	Descrição
wbmp	verdadeiro/falso	Suporta o formato wbmp.
bmp	verdadeiro/falso	Suporta o formato bmp.
epoc_bmp	verdadeiro/falso	Suporta o formato bitmap EPOC (Symbian), também conhecido como mbm.
gif	verdadeiro/falso	Suporta o formato gif.
gif_animated	verdadeiro/falso	Suporta o formato gif89a (gif's animados).
jpg	verdadeiro/falso	Suporta o formato jpg.
png	verdadeiro/falso	Suporta o formato png.
tiff	verdadeiro/falso	Suporta o formato tif.
transparent_png_alpha	verdadeiro/falso	
transparent_png_index	verdadeiro/falso	
flash_lite	verdadeiro/falso	(Obsoleto. Ver grupo flashlite)
flash_lite_1_1	verdadeiro/falso	(Obsoleto. Ver grupo flashlite)
flash_lite_download_limit	Qualquer número inteiro	(Obsoleto. Ver grupo flashlite)
svgt_1_1	verdadeiro/falso	Suporta SVGT v1.1
svgt_1_1_plus	verdadeiro/falso	Suporta SVGT v1.1+
greyscale	verdadeiro/falso	Suporta o formato greyscale.
colors	Qualquer número inteiro	Número de cores suportado pelo dispositivo.

Grupo: **bugs**

<i>Nome da capacidade</i>	Tipo	Descrição
post_method_support	verdadeiro/falso	Se verdadeiro o dispositivo suporta o método HTTP POST.
basic_authentication_support	verdadeiro/falso	Se verdadeiro o dispositivo suporta suporte de autenticação básico (login e password).
emptyok	verdadeiro/falso	É permitido o uso de empty select.
empty_option_value_support	verdadeiro/falso	Se verdadeiro o dispositivo permite ao utilizador escolher um valor "empty" a partir de um "select".

Grupo: **wta**

<i>Nome da capacidade</i>	Tipo	Descrição
nokia_voice_call	verdadeiro/falso	Suporta a função Nokia 'efectuar chamada'.
wta_voice_call	verdadeiro/falso	Suporta a função standard de WML de chamada.
wta_phonebook	verdadeiro/falso	A implementação WTA suporta acesso à lista de contactos do dispositivo.
wta_misc	verdadeiro/falso	A implementação WTA suporta os diversos recursos da especificação WTAI.
wta_pdc	verdadeiro/falso	Suporta WTA dentro de uma rede PDC.

Grupo: **security**

<i>Nome da capacidade</i>	Tipo	Descrição
https_support	supported, not_supported, not_predictable	Suporte genérico para o protocolo HTTPS (conexões SSL). A predefinição é "supported", devido à recomendação de questionar o utilizador para activar uma conexão HTTPS.
https_verisign_class3	verdadeiro/falso	Suporte para conexões SSL com certificado Verisign Class 3.
https_detectable (Obsoleto)	verdadeiro/falso	O dispositivo é capaz de detectar quando é feito um pedido a um recurso HTTPS.
phone_id_provided	verdadeiro/falso	O número de IMEI está disponível.

Grupo: **bearer**

<i>Nome da capacidade</i>	Tipo	Descrição
max_data_rate	valor em Kilobits.um kilobit = 1000 bit	Largura de banda máxima atingível pelo dispositivo. Valores possíveis: HSDPA = 1800 3600 7200 14400 dependendo do dispositivo UMTS(3G) = 384 EGPRS/EDGE = 200 GPRS = 40 HSCSD = 29 CSD = 9
wifi	verdadeiro/falso	Dispositivo consegue aceder a conexões WiFi.
sdio	verdadeiro/falso	Dispositivo aceita cartões SDIO (para WiFi).
vpn	verdadeiro/falso	Dispositivo suporta conexões VPN.

Grupo: **storage**

<i>Nome da capacidade</i>	Tipo	Descrição
max_deck_size	Qualquer número inteiro	Tamanho máximo permitido para um deck em bytes.
max_url_length_in_requests	Qualquer número inteiro	Tamanho máximo permitido do comprimento de um URL.
max_url_length_homepage	Qualquer número inteiro	Tamanho máximo permitido do comprimento do URL para a homepage do browser.
max_url_length_bookmark	Qualquer número inteiro	Tamanho máximo permitido para o comprimento do URL de um bookmark.
max_url_length_cached_page	Qualquer número inteiro	Tamanho máximo permitido do comprimento do URL para uma página em cache.
max_no_of_connection_settings	Qualquer número inteiro	Número de perfis de ligações suportado.
max_no_of_bookmarks	Qualquer número inteiro	Número máximo de bookmarks que o browser consegue armazenar.
max_length_of_username	Qualquer número inteiro	Comprimento máximo permitido para um username.
max_length_of_password	Qualquer número inteiro	Comprimento máximo permitido para uma password.
max_object_size	Qualquer número inteiro	O tamanho máximo de um ficheiro (em bytes) quando descarregado através de WTP-SAR.

Grupo: **object_download**

<i>Nome da capacidade</i>	Tipo	Descrição
downloadfun_support	verdadeiro/falso	Se verdadeiro o dispositivo suporta recursos downloadfun.
directdownload_support	verdadeiro/falso	Se verdadeiro o dispositivo suporta downloads por via de uma anchor.
inline_support	verdadeiro/falso	Se verdadeiro o dispositivo tem a possibilidade de armazenar uma imagem ou objecto visualizados numa página.
oma_support	verdadeiro/falso	Se verdadeiro o dispositivo suporta as especificações OMA para download de objectos.
ringtone	verdadeiro/falso	Se verdadeiro o dispositivo suporta download de tons de toque.
ringtone_midi_monophonic	verdadeiro/falso	Suporta download de ficheiros Midi monophonic (tipo 0).
ringtone_midi_polyphonic	verdadeiro/falso	Suporta download de ficheiros Midi polifónicos.
ringtone_imelody	verdadeiro/falso	Suporta download de ficheiros iMelody.
ringtone_digiplug	verdadeiro/falso	Suporta download de ficheiros digiplug.
ringtone_compactmidi	verdadeiro/falso	Suporta download de ficheiros compact-midi.
ringtone_mmf	verdadeiro/falso	Suporta download de ficheiros MMF/SMAF (Yamaha).
ringtone_rmf	verdadeiro/falso	Suporta download de ficheiros

		RMF (Beatnik).
ringtone_xmf	verdadeiro/falso	Suporta download de ficheiros XMF (Beatink - midi approved).
ringtone_amr	verdadeiro/falso	Suporta download de ficheiros AMR.
ringtone_awb	verdadeiro/falso	Suporta download de ficheiros de banda larga AMR.
ringtone_aac	verdadeiro/falso	Suporta download de ficheiros AAC.
ringtone_wav	verdadeiro/falso	Suporta download de ficheiros WAV .
ringtone_mp3	verdadeiro/falso	Suporta download de ficheiros MP3.
ringtone_spmidi	verdadeiro/falso	Suporta download de ficheiros SPmidi.
ringtone_voices	Qualquer número inteiro	Representa o número máximo de vozes para um tom de toque descarregado.
ringtone_df_size_limit	Qualquer número inteiro	Limite (em bytes) do tamanho de tons de toque descarregados via downloadfun.
ringtone_directdownload_size_limit	Qualquer número inteiro	Limite (em bytes) do tamanho de tons de toque descarregados via download directo.
ringtone_inline_size_limit	Qualquer número inteiro	Limite (em bytes) do tamanho de tons de toque descarregados via inline objects
ringtone_oma_size_limit	Qualquer	Limite (em bytes) do tamanho

	número inteiro	de tons de toque descarregados via OMA DD.
wallpaper	verdadeiro/falso	Se verdadeiro o dispositivo suporta download de wallpapers.
wallpaper_wbmp	verdadeiro/falso	Suporte para imagens wbmp.
wallpaper_bmp	verdadeiro/falso	Suporte para imagens bmp.
wallpaper_gif	verdadeiro/falso	Suporte para imagens gif images
wallpaper_jpg	verdadeiro/falso	Suporte para imagens jpg.
wallpaper_png	verdadeiro/falso	Suporte para imagens png.
wallpaper_greyscale	verdadeiro/falso	Verdadeiro se o dispositivo usa greyscale.
wallpaper_colors	Qualquer número inteiro	Número de cores disponíveis. (em bits). Nota: se um dispositivo usa 8 tons de cinzento, a capacidade wallpaper_greyscale deverá ser marcada como <i>true</i> e a capacidade wallpaper_colors para 3.
wallpaper_max_width	Qualquer número inteiro	Largura máxima suportada por um wallpaper.
wallpaper_max_height	Qualquer número inteiro	Altura máxima suportada por um wallpaper.
wallpaper_preferred_width	Qualquer número inteiro	Largura máxima sugerida para um wallpaper.
wallpaper_preferred_height	Qualquer número inteiro	Altura máxima sugerida para um wallpaper.
wallpaper_df_size_limit	Qualquer número inteiro	Tamanho máximo (em bytes) de um wallpaper descarregado por downloadfun.

wallpaper_directdownload_size_limit	Qualquer número inteiro	Tamanho máximo (em bytes) de um wallpaper descarregado por download directo.
wallpaper_inline_size_limit	Qualquer número inteiro	Tamanho máximo (em bytes) de um wallpaper descarregado como inline object.
wallpaper_oma_size_limit	Qualquer número inteiro	Tamanho máximo (em bytes) de um wallpaper descarregado via OMA DD.
wallpaper_resize	string	Descreve a capacidade de o dispositivo efectuar resizing de wallpapers descarregados no caso de não terem o tamanho exacto do visor. Valores possíveis: "none", "fixed_ratio" (reescalonamento respeitando as proporções originais), "crop_centered", "crop_top_left".
screensaver	verdadeiro/falso	Se verdadeiro o dispositivo suporta o download de screensavers
screensaver_wbmp	verdadeiro/falso	Suporte para imagens wbmp.
screensaver_bmp	verdadeiro/falso	Suporte para imagens bmp.
screensaver_gif	verdadeiro/falso	Suporte para imagens gif.
screensaver_jpg	verdadeiro/falso	Suporte para imagens jpg.
screensaver_png	verdadeiro/falso	support for png images
screensaver_greyscale	verdadeiro/falso	true if the phone users a greyscale
screensaver_colors	Qualquer número inteiro	Número de cores disponíveis. (em bits). Nota: se um dispositivo usa 8 tons de cinzento, a capacidade

		screensaver_greyscale deverá ser marcada como <i>true</i> e a capacidade screensaver_colors para 3.
screensaver_max_width	Qualquer número inteiro	Largura máxima suportada para um screensaver.
screensaver_max_height	Qualquer número inteiro	Altura máxima suportada para um screensaver.
screensaver_preferred_width	Qualquer número inteiro	Largura máxima sugerida para um screensaver.
screensaver_preferred_height	Qualquer número inteiro	Altura máxima sugerida para um screensaver.
screensaver_df_size_limit	Qualquer número inteiro	Tamanho máximo (em bytes) de um screensaver descarregado por downloadfun.
screensaver_directdownload_size_limit	Qualquer número inteiro	Tamanho máximo (em bytes) de um wallpaper descarregado por download directo.
screensaver_inline_size_limit	Qualquer número inteiro	Tamanho máximo (em bytes) de um screensaver descarregado como inline object.
screensaver_oma_size_limit	Qualquer número inteiro	Tamanho máximo (em bytes) de um screensaver descarregado por OMA DD.
screensaver_resize	string	Descreve a capacidade de o dispositivo efectuar resizing de screensavers descarregados no caso de não terem o tamanho exacto do visor. Valores possíveis: "none", "fixed_ratio" (reescalonamento respeitando as proporções

		originais), "crop_centered", "crop_top_left".
picture	verdadeiro/falso	Se verdadeiro o dispositivo suporta o download de ficheiros picture.
picture_wbmp	verdadeiro/falso	Suporte para imagens wbmp.
picture_bmp	verdadeiro/falso	Suporte para imagens bmp
picture_gif	verdadeiro/falso	Suporte para imagens gif.
picture_jpg	verdadeiro/falso	Suporte para imagens jpg.
picture_png	verdadeiro/falso	Suporte para imagens png.
picture_greyscale	verdadeiro/falso	Verdadeiro se o dispositivo usa greyscale.
picture_colors	Qualquer número inteiro	Número de cores disponíveis. (em bits). Nota: se um dispositivo usa 8 tons de cinzento, a capacidade picture_greyscale deverá ser marcada como <i>true</i> e a capacidade picture_colors para 3.
picture_max_width	Qualquer número inteiro	Largura máxima suportada para uma picture.
picture_max_height	Qualquer número inteiro	Altura máxima suportada para uma picture.
picture_preferred_width	Qualquer número inteiro	Largura máxima sugerida para uma picture.
picture_preferred_height	Qualquer número inteiro	Altura máxima sugerida para uma picture.
picture_df_size_limit	Qualquer número inteiro	Tamanho máximo (em bytes) de uma picture descarregada por downloadfun.

picture_directdownload_size_limit	Qualquer número inteiro	Tamanho máximo (em bytes) de uma picture descarregada por download directo.
picture_inline_size_limit	Qualquer número inteiro	Tamanho máximo (em bytes) de uma picture descarregada como inline object.
picture_oma_size_limit	Qualquer número inteiro	Tamanho máximo (em bytes) de uma picture descarregada por OMA DD.
picture_resize	string	Descreve a capacidade de o dispositivo efectuar resizing de pictures descarregadas no caso de não terem o tamanho exacto do visor. Valores possíveis: "none", "fixed_ratio" (reescalonamento respeitando as proporções originais), "crop_centered", "crop_top_left".
video	verdadeiro/falso	Se verdadeiro, o dispositivo suporta downloads de vídeos.
video_real_media_8	verdadeiro/falso	Se verdadeiro, o dispositivo suporta real media V8.
video_real_media_9	verdadeiro/falso	Se verdadeiro, o dispositivo suporta real media V9.
video_real_media_10	verdadeiro/falso	Se verdadeiro, o dispositivo suporta real media V10.
video_3gpp	verdadeiro/falso	Se verdadeiro, o dispositivo suporta vídeos 3gpp. (incluindo H.263)
video_3gpp2	verdadeiro/falso	Se verdadeiro, o dispositivo suporta vídeos 3gpp2. (para dispositivos CDMA)

video_mp4	verdadeiro/falso	Se verdadeiro, o dispositivo suporta vídeos mp4.
video_wmv	verdadeiro/falso	Se verdadeiro, o dispositivo suporta vídeos WMV.
video_mov	verdadeiro/falso	Se verdadeiro, o dispositivo suporta vídeos MOV.
video_max_frame_rate	inteiro	Frame rate máxima suportada. A predefinição é 0.
video_max_width	inteiro	Largura máxima do vídeo. A predefinição é 0.
video_max_height	inteiro	Altura máxima do vídeo. A predefinição é 0.
video_qcif	verdadeiro/falso	Se verdadeiro, o dispositivo suporta vídeos QCIF.
video_sqcif	verdadeiro/falso	Se verdadeiro, o dispositivo suporta vídeos SQIF.
video_preferred_width	inteiro	Altura máxima recomendada de modo a que o vídeo não necessite resizing . A predefinição é 0.
video_preferred_height	inteiro	Largura máxima recomendada de modo a que o vídeo não necessite resizing . A predefinição é 0.
video_df_size_limit	inteiro	Tamanho máximo (em bytes) para vídeos descarregados por downloadfun. A predefinição é 0.
video_directdownload_size_limit	inteiro	Tamanho máximo (em bytes) para vídeos descarregados por download directo. A predefinição é 0.

video_inline_size_limit	inteiro	Tamanho máximo (em bytes) para vídeos descarregados por inline download. A predefinição é 0.
video_oma_size_limit	inteiro	Tamanho máximo (em bytes) para vídeos descarregados por OMA DD. A predefinição é 0.
video_vcodec_h263_0	verdadeiro/falso	Se verdadeiro, o dispositivo consegue reproduzir vídeos codificados por H.263 versão 0.
video_vcodec_h263_3	verdadeiro/falso	Se verdadeiro, o dispositivo consegue reproduzir vídeos codificados por H.263 versão 3.
video_vcodec_h264	String	none = sem suporte H264. A string representa os diferentes níveis de perfil de baseline. Ex: "10 1b", "10 1b 11 12"
video_vcodec_mpeg4	verdadeiro/falso	Se verdadeiro, o dispositivo consegue reproduzir vídeos codificados por MPEG 4.
video_acodec_amr	verdadeiro/falso	Se verdadeiro, o dispositivo consegue reproduzir vídeos com áudio AMR.
video_acodec_awb	verdadeiro/falso	Se verdadeiro, o dispositivo consegue reproduzir vídeos com áudio de banda larga AMR.
video_acodec_aac	verdadeiro/falso	Se verdadeiro, o dispositivo consegue reproduzir vídeos com áudio AAC.
video_acodec_aac_ltp	verdadeiro/falso	Se verdadeiro, o dispositivo consegue reproduzir vídeos

		com áudio AAC LTP.
video_acodec_qcelp	verdadeiro/falso	Se verdadeiro, o dispositivo consegue reproduzir vídeos com formato áudio Qualcomm Code Excited Linear Predictive waveform.

Grupo: **drm**

<i>Nome da capacidade</i>	Tipo	Descrição
oma_v_1_0_forwardlock	verdadeiro/falso	Se verdadeiro, o dispositivo suporta OMA DRM ForwardLock V1.0.
oma_v_1_0_combined_delivery	verdadeiro/falso	Se verdadeiro, o dispositivo suporta OMA DRM Combined Delivery V1.0.
oma_v_1_0_separate_delivery	verdadeiro/falso	Se verdadeiro, o dispositivo suporta OMA DRM Separate Delivery V1.0.

Grupo: **streaming**

<i>Nome da capacidade</i>	Tipo	Descrição
streaming_video	verdadeiro/falso	Se verdadeiro, o dispositivo suporta streaming vídeo.
streaming_real_media_8	verdadeiro/falso	Se verdadeiro, o dispositivo suporta streaming vídeo Real media V8.
streaming_real_media_9	verdadeiro/falso	Se verdadeiro, o dispositivo suporta streaming vídeo Real media V9.
streaming_real_media_10	verdadeiro/falso	Se verdadeiro, o dispositivo suporta streaming vídeo Real media V10.
streaming_3gpp	verdadeiro/falso	Se verdadeiro, o dispositivo suporta streaming vídeo 3GPP.
streaming_mp4	verdadeiro/falso	Se verdadeiro, o dispositivo suporta streaming vídeo MP4.
streaming_wmv	verdadeiro/falso	Se verdadeiro, o dispositivo suporta streaming vídeo WMV.
streaming_mov	verdadeiro/falso	Se verdadeiro, o dispositivo suporta streaming vídeo MOV.
streaming_video_qcif	verdadeiro/falso	Se verdadeiro, o dispositivo suporta streaming vídeo QCIF standard.
streaming_video_qcif_max_width	inteiro	Largura máxima recomendada para vídeos QCIF.
streaming_video_qcif_max_height	inteiro	Altura máxima recomendada para vídeos QCIF.
streaming_video_sqcif	verdadeiro/falso	Se verdadeiro, o dispositivo suporta streaming vídeo SQCIF standard.

streaming_video_sqcif_max_width	inteiro	Largura máxima recomendada para vídeos SQCIF .
streaming_video_sqcif_max_height	inteiro	Altura máxima recomendada para vídeos SQCIF.
streaming_video_max_bit_rate	inteiro	Débito binário máximo para vídeo+áudio.
streaming_video_max_video_bit_rate	inteiro	Débito binário máximo para vídeo.
streaming_video_min_video_bit_rate	inteiro	Débito binário mínimo para vídeo.
streaming_video_max_audio_bit_rate	inteiro	Débito binário máximo para áudio
.streaming_video_max_frame_rate	inteiro	Frame rate máxima de streaming vídeo.
streaming_video_size_limit	inteiro	Tamanho máximo (em bytes) do clip vídeo streaming.
streaming_video_vcodec_h263_0	verdadeiro/falso	Se verdadeiro, o dispositivo suporta vídeos streaming codificados com H.263 tipo 0.
streaming_video_vcodec_h263_3	verdadeiro/falso	Se verdadeiro, o dispositivo suporta vídeos streaming codificados com H.263 tipo 3.
streaming_video_vcodec_mpeg4	verdadeiro/falso	Se verdadeiro, o dispositivo suporta vídeos streaming codificados com MPEG 4.
streaming_video_vcodec_h264	verdadeiro/falso	Se verdadeiro, o dispositivo suporta vídeos streaming codificados com H264 (this is the streaming counterpart of (object download) video_vcodec_h264
streaming_video_acodec_amr	verdadeiro/falso	Se verdadeiro, o dispositivo suporta vídeos streaming com

		codec áudio AMR.
streaming_video_acodec_awb	verdadeiro/falso	Se verdadeiro, o dispositivo suporta vídeos streaming com codec áudio AWB.
streaming_video_acodec_aac	verdadeiro/falso	Se verdadeiro, o dispositivo suporta vídeos streaming com codec áudio AAC.
streaming_video_acodec_aac_ltp	verdadeiro/falso	Se verdadeiro, o dispositivo suporta vídeos streaming com codec áudio AAC LTP.

Grupo: **wap_push**

<i>Nome da capacidade</i>	Tipo	Descrição
wap_push_support	verdadeiro/falso	Se verdadeiro, o dispositivo suporta mensagens WAP push.
connectionless_service_indication	verdadeiro/falso	Se verdadeiro, o dispositivo suporta indicação do serviço connectionless.
connectionless_service_load	verdadeiro/falso	Se verdadeiro, o dispositivo suporta operações de carga connectionless.
connectionless_cache_operation	verdadeiro/falso	Se verdadeiro, o dispositivo suporta operações de cache.
connectionoriented_unconfirmed_service_indication	verdadeiro/falso	Se existe suporte para indicações de serviço não confirmadas quando é usado o serviço push connection-oriented.
connectionoriented_unconfirmed_service_load	verdadeiro/falso	Se existe suporte para indicações de serviço de carga não confirmadas quando é usado o serviço push connection-oriented.
connectionoriented_unconfirmed_cache_operation	verdadeiro/falso	Se existe suporte para operações de cache não confirmadas quando é usado o serviço push connection-oriented.
connectionoriented_confirmed_service_indication	verdadeiro/falso	Se existe suporte para indicações de serviço confirmadas quando é usado o serviço push connection-oriented.
connectionoriented_confirmed_service_load	verdadeiro/falso	Se existe suporte para indicações de serviço de carga confirmadas quando é usado o serviço push connection-oriented.
connectionoriented_confirmed_	verdadeiro/falso	Se existe suporte para operações de

cache_operation		cache confirmadas quando é usado o serviço push connection-oriented.
utf8_support	verdadeiro/falso	Se é suportado o character set UTF-8.
ascii_support	verdadeiro/falso	Se é suportado o character set ASCII.
iso8859_support	verdadeiro/falso	Se é suportado o character set ISO-8859.
expiration_date	verdadeiro/falso	Se pode ser criada uma data de expiração do serviço.

Grupo: **mms**

<i>Nome da capacidade</i>	Tipo	Descrição
receiver	verdadeiro/falso	Se pode receber mensagens MMS.
sender	verdadeiro/falso	Se pode enviar mensagens MMS.
mms_max_height	Qualquer número inteiro	Altura máxima de uma imagem numa mensagem MMS.
mms_max_width	Qualquer número inteiro	Largura máxima de uma imagem numa mensagem MMS.
built_in_recorder	verdadeiro/falso	Se o dispositivo possui gravador de áudio.
built_in_camera	verdadeiro/falso	Se o dispositivo possui câmara.
mms_jpeg_baseline	verdadeiro/falso	Se o dispositivo possui suporte para imagens Baseline JPG em mensagens MMS.
mms_jpeg_progressive	verdadeiro/falso	Se o dispositivo possui suporte para imagens Progressive JPG em mensagens MMS.
mms_gif_static	verdadeiro/falso	Se o dispositivo possui suporte para imagens Static GIF (87a) em mensagens MMS.
mms_gif_animated	verdadeiro/falso	Se o dispositivo possui suporte para imagens animadas GIF (87a) em mensagens MMS.
mms_png	verdadeiro/falso	Se o dispositivo possui suporte para imagens PNG em mensagens MMS.
mms_bmp	verdadeiro/falso	Se o dispositivo possui suporte para imagens BMP em mensagens MMS.
mms_wbmp	verdadeiro/falso	Se o dispositivo possui suporte para imagens WBMP em mensagens MMS.

mms_amr	verdadeiro/falso	Se o dispositivo possui suporte áudio AMR em mensagens MMS.
mms_wav	verdadeiro/falso	Se o dispositivo possui suporte áudio WAV em mensagens MMS.
mms_midi_monophonic	verdadeiro/falso	Se o dispositivo possui suporte áudio Monophonic MIDI em mensagens MMS.
mms_midi_polyphonic	verdadeiro/falso	Se o dispositivo possui suporte áudio Polyphonic MIDI em mensagens MMS.
mms_midi_polyphonic_voices	inteiro	Em caso de suportar Polyphonic MIDI, o número de vozes suportado.
mms_spmidi	verdadeiro/falso	Se o dispositivo possui suporte áudio SPMIDI em mensagens MMS.
mms_ota_bitmap	verdadeiro/falso	Se o dispositivo possui suporte OTA Bitmap em mensagens MMS.
mms_nokia_wallpaper	verdadeiro/falso	Se o dispositivo possui suporte Nokia wallpaper em mensagens MMS.
mms_nokia_operatorlogo	verdadeiro/falso	Se o dispositivo possui suporte Nokia operator logo em mensagens MMS.
mms_nokia_3dscreensaver	verdadeiro/falso	Se o dispositivo possui suporte Nokia 3D screensaver em mensagens MMS.
mms_nokia_ringtone	verdadeiro/falso	Se o dispositivo possui suporte Nokia ringtone.
mms_rmf	verdadeiro/falso	Se o dispositivo possui suporte RMF.
mms_symbian_install	verdadeiro/falso	Suporte para instalação de ficheiros Symbian. (Pode receber ficheiros Symbian anexados numa MMS.)
mms_jar	verdadeiro/falso	Suporte para instalação de ficheiros JAR. (Pode receber ficheiros JAR anexados numa MMS.)
mms_jad	verdadeiro/falso	Suporte para instalação de ficheiros

		JAD. (Pode receber ficheiros JAD anexados numa MMS.)
mms_vcard	verdadeiro/falso	Suporte para Vcard.
mms_wml	verdadeiro/falso	A mensagem pode incluir wml.
mms_wbxml	verdadeiro/falso	A mensagem pode incluir wbxml.
mms_wmlc	verdadeiro/falso	A mensagem pode incluir wbxml.
mms_video	verdadeiro/falso	A mensagem pode incluir um clip de vídeo.
mms_mp4	verdadeiro/falso	A mensagem pode incluir um clip de vídeo MP4.
mms_3gpp	verdadeiro/falso	A mensagem pode incluir um clip de vídeo 3GPP.
mms_3gpp2	verdadeiro/falso	A mensagem pode incluir um clip de vídeo 3GPP2. (Para dispositivos CDMA)
mms_max_frame_rate	inteiro	Frame rate máxima de um clip de vídeo incluído numa MMS.

Grupo: **sms** SMS Binário e capacidades SCKL.

<i>Nome da capacidade</i>	Tipo	Descrição
nokiaring	verdadeiro/falso	Se verdadeiro, o dispositivo suporta nokiaring.
picturemessage	verdadeiro/falso	Se verdadeiro, o dispositivo suporta Nokia picture messages.
operatorlogo	verdadeiro/falso	Se verdadeiro, o dispositivo suporta Nokia operator logo's (72x14 pixel).
largeoperatorlogo	verdadeiro/falso	Se verdadeiro, o dispositivo suporta Nokia large operator logo's (72x28 pixel).
callericon	verdadeiro/falso	Se verdadeiro, o dispositivo suporta Nokia caller icons.
nokiavcard	verdadeiro/falso	Se verdadeiro, o dispositivo suporta Nokia vcards.
nokiavcal	verdadeiro/falso	Se verdadeiro, o dispositivo suporta Nokia vcal.
sckl_ringtone	verdadeiro/falso	Se verdadeiro, o dispositivo suporta SCKL ringtones.
sckl_operatorlogo	verdadeiro/falso	Se verdadeiro, o dispositivo suporta SCKL operator logos.
sckl_groupgraphic	verdadeiro/falso	Se verdadeiro, o dispositivo suporta SCKL group graphics.
sckl_vcard	verdadeiro/falso	Se verdadeiro, o dispositivo suporta SCKL vcards.
sckl_vcal	verdadeiro/falso	Se verdadeiro, o dispositivo suporta SCKL vcal.
text_imelody	verdadeiro/falso	Se verdadeiro, o dispositivo suporta textual iMelody.
ems	verdadeiro/falso	Se verdadeiro, o dispositivo suporta EMS

		messages.
ems_variablesizedpictures	verdadeiro/falso	Se verdadeiro, o dispositivo suporta EMS messages com pictures de tamanho variável.
ems_imelody	verdadeiro/falso	Se verdadeiro, o dispositivo suporta iMelody over EMS messages.
ems_odi	verdadeiro/falso	Se verdadeiro, o dispositivo suporta EMS ODI (Object Distribution Indicator).
ems_upi	verdadeiro/falso	Se verdadeiro, o dispositivo suporta EMS UPI (User Prompt Indicator).
ems_version	inteiro	Versão EMS.
siemens_ota	verdadeiro/falso	Se verdadeiro, o dispositivo suporta Siemens OTA.
siemens_logo_width	inteiro	Largura do logo (predefinição é 101).
siemens_logo_height	inteiro	Altura do logo height (predefinição é 29).
siemens_screensaver_width	inteiro	Largura screensaver (predefinição é 101).
siemens_screensaver_height	inteiro	Altura screensaver (predefinição é 50).
gprtf	verdadeiro/falso	Se verdadeiro, o dispositivo suporta Motorola proprietary ringtones (GPRTF).
sagem_v1	verdadeiro/falso	Se verdadeiro, o dispositivo suporta proprietary ringtones spec 1.0.
sagem_v2	verdadeiro/falso	Se verdadeiro, o dispositivo suporta Sagem proprietary ringtones spec 2.0.
panasonic	verdadeiro/falso	Se verdadeiro, o dispositivo suporta Panasonic proprietary ringtones.

Grupo: **j2me**

<i>Nome da capacidade</i>	Tipo	Descrição
j2me_midp_1_0	verdadeiro/falso	Se verdadeiro, o dispositivo cumpre a especificação MIDP 1.0
j2me_cldc_1_0	verdadeiro/falso	Se verdadeiro, o dispositivo cumpre a especificação CLDC 1.0.
j2me_midp_2_0	verdadeiro/falso	Se verdadeiro, o dispositivo cumpre a especificação MIDP 2.0
j2me_cldc_1_1	verdadeiro/falso	Se verdadeiro, o dispositivo cumpre a especificação CLDC 1.1
doja_1_0	verdadeiro/falso	Se verdadeiro, o dispositivo cumpre a especificação DoJa 1.0
doja_1_5	verdadeiro/falso	Se verdadeiro, o dispositivo cumpre a especificação DoJa 1.5
doja_2_0	verdadeiro/falso	Se verdadeiro, o dispositivo cumpre a especificação DoJa 2.0
doja_2_1	verdadeiro/falso	Se verdadeiro, o dispositivo cumpre a especificação DoJa 2.1
doja_2_2	verdadeiro/falso	Se verdadeiro, o dispositivo cumpre a especificação DoJa 2.2
doja_3_0	verdadeiro/falso	Se verdadeiro, o dispositivo cumpre a especificação DoJa 3.0
doja_3_5	verdadeiro/falso	Se verdadeiro, o dispositivo cumpre a especificação DoJa 3.5
doja_4_0	verdadeiro/falso	Se verdadeiro, o dispositivo cumpre a especificação DoJa 4.0

j2me_jtwi	verdadeiro/falso	Suporte para JTWI.
j2me_mmapi_1_0	verdadeiro/falso	Suporte para MMAPI 1.0
j2me_mmapi_1_1	verdadeiro/falso	Suporte para MMAPI 1.1
j2me_wmapi_1_0	verdadeiro/falso	Suporte para WMAPI 1.0
j2me_wmapi_1_1	verdadeiro/falso	Suporte para WMAPI 1.1
j2me_wmapi_2_0	verdadeiro/falso	Suporte para WMAPI 2.0
j2me_btapi	verdadeiro/falso	Suporte para a API BlueTooth.
j2me_3dapi	verdadeiro/falso	Suporte para a API 3D.
j2me_locapi	verdadeiro/falso	Suporte para a API Location.
j2me_nokia_ui	verdadeiro/falso	Suporte para Nokia UI.
j2me_motorola_lwt	verdadeiro/falso	Suporte para Motorola LWT.
j2me_siemens_color_game	verdadeiro/falso	Suporte para Siemens Color Game extension.
j2me_siemens_extension	verdadeiro/falso	Suporte para Siemens extension.
j2me_heap_size	any integer number	Limite da memória (em bytes) em runtime.
j2me_storage_size	any integer number	Limite da memória física.
j2me_max_jar_size	any integer number	Limite (em bytes) de um midlet descarregado OTA.
j2me_max_record_store_size	any integer number	Limite (em bytes) de um record store.
j2me_screen_width	any integer number	Largura do visor em pixels.
j2me_screen_height	any integer number	Altura do visor em pixels.
j2me_canvas_width	any integer number	Largura do canvas em pixels.

j2me_canvas_height	any integer number	Altura do canvas em pixels.
j2me_bits_per_pixel	any integer number	Bits por pixels – indica as cores suportadas.
j2me_audio_capture_enabled	verdadeiro/falso	O dispositivo consegue gravar clips áudio durante a execução de um midlet.
j2me_video_capture_enabled	verdadeiro/falso	O dispositivo consegue gravar clips vídeo durante a execução de um midlet.
j2me_photo_capture_enabled	verdadeiro/falso	O dispositivo consegue capturar imagens durante a execução de um midlet.
j2me_capture_image_formats	string	Se a capacidade anterior for verdadeira, indica o formato suportado da imagem.
j2me_http	verdadeiro/falso	O dispositivo consegue iniciar sessões http.
j2me_https	verdadeiro/falso	O dispositivo consegue iniciar sessões https.
j2me_socket	verdadeiro/falso	O dispositivo consegue abrir um socket.
j2me_udp	verdadeiro/falso	O dispositivo consegue iniciar sessões udp.
j2me_serial	verdadeiro/falso	O dispositivo consegue iniciar ligações em série.
j2me_gif	verdadeiro/falso	Suporte para imagens gif.
j2me_gif89a	verdadeiro/falso	Suporte para imagens animadas gif 89a.
j2me_jpg	verdadeiro/falso	Suporte para imagens jpg.
j2me_png	verdadeiro/falso	Suporte para imagens png.

j2me_bmp	verdadeiro/falso	Suporte para imagens bmp.
j2me_bmp3	verdadeiro/falso	Suporte para imagens bmp3.
j2me_wbmp	verdadeiro/falso	Suporte para imagens wbmp.
j2me_midi	verdadeiro/falso	Suporte para ficheiros Midi.
j2me_wav	verdadeiro/falso	Suporte para ficheiros wav.
j2me_amr	verdadeiro/falso	Suporte para ficheiros amr.
j2me_mp3	verdadeiro/falso	Suporte para ficheiros mp3.
j2me_mp4	verdadeiro/falso	Suporte para ficheiros mp4.
j2me_imelody	verdadeiro/falso	Suporte para ficheiros imelody.
j2me_rmf	verdadeiro/falso	Suporte para ficheiros rmf .
j2me_au	verdadeiro/falso	Suporte para ficheiros au.
j2me_aac	verdadeiro/falso	Suporte para ficheiros aac .
j2me_realaudio	verdadeiro/falso	Suporte para ficheiros realaudio.
j2me_xmf	verdadeiro/falso	Suporte para ficheiros xmf.
j2me_wma	verdadeiro/falso	Suporte para ficheiros wma.
j2me_3gpp	verdadeiro/falso	Suporte para ficheiros 3gpp.
j2me_h263	verdadeiro/falso	Suporte para ficheiros h263.
j2me_svgt	verdadeiro/falso	Suporte para ficheiros svgt.
j2me_mpeg4	verdadeiro/falso	Suporte para ficheiros mpeg4 áudio/vídeo.
j2me_realvideo	verdadeiro/falso	Suporte para ficheiros realvideo.
j2me_real8	verdadeiro/falso	Suporte para ficheiros real8.
j2me_realmedia	verdadeiro/falso	Suporte para ficheiros realmedia.
j2me_left_softkey_code	Qualquer número inteiro	O número devolvido por esta capacidade refere-se ao valor retornado por um método da

		<p>classe event handler da API JME que depende da implementação efectuada no dispositivo. É o valor retornado quando é pressionada a tecla softkey esquerda. (geralmente é um número negativo).</p>
j2me_right_softkey_code	Qualquer número inteiro	<p>O número devolvido por esta capacidade refere-se ao valor retornado por um método da classe event handler da API JME que depende da implementação efectuada no dispositivo. É o valor retornado quando é pressionada a tecla softkey direita. (geralmente é um número negativo).</p>
j2me_middle_softkey_code	Qualquer número inteiro	<p>O número devolvido por esta capacidade refere-se ao valor retornado por um método da classe event handler da API JME que depende da implementação efectuada no dispositivo. É o valor retornado quando é pressionada a tecla softkey central. (geralmente é um número negativo).</p>
j2me_select_key_code	Qualquer número inteiro	<p>O número devolvido por esta capacidade refere-se ao valor retornado por um método da classe event handler da API JME que depende da implementação efectuada no dispositivo. É o valor retornado quando é pressionada a tecla 'select' softkey. (geralmente é um número negativo).</p>

j2me_return_key_code	Qualquer número inteiro	O número devolvido por esta capacidade refere-se ao valor retornado por um método da classe event handler da API JME que depende da implementação efectuada no dispositivo. É o valor retornado quando é pressionada a tecla 'return' softkey. (geralmente é um número negativo).
j2me_clear_key_code	Qualquer número inteiro	O número devolvido por esta capacidade refere-se ao valor retornado por um método da classe event handler da API JME que depende da implementação efectuada no dispositivo. É o valor retornado quando é pressionada a tecla 'clear' softkey. (geralmente é um número negativo).
j2me_datefield_no_accepts_null_date	verdadeiro/falso	Se o campo da data aceita ou não valores nulos. (referenciado como bug)
j2me_datefield_broken	verdadeiro/falso	Se o campo da data funciona ou não. (referenciado como bug)

Grupo: **sound_format** (Formatos de som suportados)

<i>Nome da capacidade</i>	Tipo	Descrição
wav	verdadeiro/falso	Suporta o formato de som wav (Waveform).
mmf	verdadeiro/falso	Suporta o formato de som MMF (um formato Yamaha). A versão é dependente do número de vozes.
smf	verdadeiro/falso	Suporta o formato de som smf (Standard MIDI File).
mld	verdadeiro/falso	Suporta o formato de som mld (formato de som para iMode).
midi_monophonic	verdadeiro/falso	Suporta o formato de som monofónico midi (Musical Instrument Digital Interface).
midi_polyphonic	verdadeiro/falso	Suporta o formato de som polifónico midi (Musical Instrument Digital Interface).
sp_midi	verdadeiro/falso	Suporta o formato de som Scalable Polyphony MIDI.
rmf	verdadeiro/falso	Suporta o formato de som rmf (formato Beatnik)
xmf	verdadeiro/falso	Suporta o formato de som XMF sound format (formato Beatnik)
compactmidi	verdadeiro/falso	Suporta o formato de som Compact MIDI (formato proprietário Faith Inc.).
digiplug	verdadeiro/falso	Suporta o formato de som compact polyphonic sound desenvolvido pela empresa Digiplug.
nokia_ringtone	verdadeiro/falso	Suporta o formato de som Nokia ringing tone.
imelody	verdadeiro/falso	Suporta o formato de som standart para melodias, também adoptado pelas 4 empresas que desenvolvem o standart EMS.
au	verdadeiro/falso	Suporta o formato de som também conhecido como uLaw, NeXT, ou Sun Audio format.
amr	verdadeiro/falso	Suporta o formato de som standart AMR.

awb	verdadeiro/falso	Suporta o formato de som de banda larga AMR.
aac	verdadeiro/falso	Suporta o formato de som standart AAC.
mp3	verdadeiro/falso	Suporta o formato de som mp3.
qcelp	verdadeiro/falso	Suporta o formato de som Qualcomm Code Excited Linear Predictive waveform.
evrc	verdadeiro/falso	Suporta o formato de som Enhanced Variable Rate Codec waveform.
voices	any integer number	Número máximo de notas que o dispositivo consegue reproduzir ao mesmo tempo.

Grupo: **flash_lite** Macromedia/Adobe Flash Lite

<i>Nome da capacidade</i>	Tipo	Descrição
flash_lite_version	1_1, 1_2, 2_0, 2_1,....	A versão de Flash Lite suportada pelo dispositivo.
fl_wallpaper	verdadeiro/falso	Aplicação Wallpaper.
fl_screensaver	verdadeiro/falso	Aplicação Screensaver.
fl_standalone	verdadeiro/falso	Standalone Flash lite.
fl_browser	verdadeiro/falso	Se o dispositivo é capaz de reproduzir ficheiros SWF files embutidos em HTML.
fl_sub_cld	verdadeiro/falso	SubLCD refere-se ao visor externo ou secundário do dispositivo e à sua capacidade de reproduzir ficheiros SWF.

Grupo: **transcoding** Tratamento de transcodificadores abusivos.

<i>Nome da capacidade</i>	Tipo	Descrição
is_transcoder	verdadeiro/falso	Transcodificadores escondem informação real do dispositivo. Esta capacidade retorna true se é detectado um transcodificador e pode ser usada para tratar este pedido específico.
transcoder_ua_header	String	Transcodificadores poderão eventualmente colocar a string de UA original num cabeçalho diferente. Esta capacidade contém o nome do cabeçalho onde a string de UA original *poderá* ser encontrada.

Group: **rss** Suporte nativo para feeds RSS

<i>Nome da capacidade</i>	Tipo	Descrição
rss_support	verdadeiro/falso	Se o dispositivo tem suporte nativo para feeds RSS.

Group: **pdf** Suporte nativo para documentos PDF

<i>Nome da capacidade</i>	Tipo	Descrição
pdf_support	verdadeiro/falso	Se o dispositivo é capaz de ler documentos pdf nativamente.

Referências

- [1] WURFL. Disponível em <http://wurfl.sourceforge.net/>. Acesso em Outubro de 2008.
- [2] Openwave Developer Network. Disponível em <http://developer.openwave.com/dvl/>. Acesso em Outubro de 2008.
- [3] “A Device-Independent System Architecture for Adaptive Mobile Learning”, Xinyou Zhao, Toshio Okamoto, 2008.
- [4] “Global Authoring Practices for the Mobile Web”. Disponível em <http://www.passani.it/gap/>. Acesso em Novembro de 2008.
- [5] “UAProf Reference”. Disponível em <http://www.developershome.com/wap/uaprofRef/>. Acesso em Outubro de 2008.
- [6] “Glossary of Terms for Device Independence”. Disponível em <http://www.w3.org/TR/di-gloss/>. Acesso em Outubro de 2008.
- [7] The Apache software foundation. Disponível em <http://www.apache.org/>. Acesso em Outubro de 2008.
- [8] “A Device-Independent System Architecture for Adaptive Mobile learning”, Aimin Liang, Suwei Guo, Choulsu Li, 2008.
- [9] “Build multiserving Java applications, Part 1”, Bilal Siddiqui, 2006.
- [10] “Build multiserving Java applications, Part 2”, Bilal Siddiqui, 2006.
- [11] “Adapting Web Content to Mobile User Agents”, Timo Laakko, Tapio Hiltunen, 2005.
- [12] “Customizing and Delivering Mobiles Services using Software Agents and CC/PP”, Qusay H. Mahmoud, Zhixin Wang, 2006.
- [13] “Software agent-based applications, Platforms and development kits”, R. Unland, M. Klusch, M. Calisti, 2000.
- [14] “UMA enabled environment for mobile media using MPEG-21 Client and Server Technology”, A. Perkis, P. Drege, O. I. Hillestad, 2005.
- [15] “An Intelligent Agent for Adapting and Delivering Electronic Course Materials to Mobile Learners”, Mohamed Ally, Fuhua Lin, Rory McGreal, Brian Woo.
- [16] “Mobile Web Design”, Cameron Moll, 2008.
- [17] “Personalized Adaptive Content System for Context-Aware Mobile Learning”, Xinyou Zhao, Fumihiko Anma, Toshie Ninomiya, Toshio Okamoto, 2008.

